AD-A063 369 BURROUGHS CORP PAOLI PA FEDERAL AND SPECIAL SYSTEMS GROUP F/G 17/2

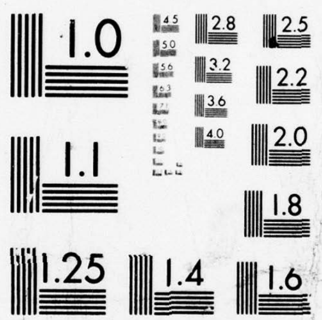
MODULAR SYSTEM CONTROL DEVELOPMENT MODEL (MSCDM). SECTIONS 1-7.(U)

SEP 77 DCA100-76-C-0083

UNCLASSIFIED 64280 SBIE-AD-E100 136 NL

1 OF 3

AD A063369



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AD-E100136

(11) LEVEL II

64280

SECTIONS 1–7

SEPTEMBER 1977

PHASE I FINAL REPORT

FOR THE
MODULAR SYSTEM CONTROL
DEVELOPMENT MODEL
(MSCDM)

for

THE DEFENSE COMMUNICATIONS AGENCY
WASHINGTON, D.C. 20305

Burroughs Corporation
Federal and Special Systems Group
Paoli, Pa. 19301

78 12 11 178

(18) FBIE

(19) AD-E100 136

(14) 64280

SECTIONS 1–7

(11) SEPTEMBER 1977

(12) 247p.

(9)

PHASE I FINAL REPORT.

Sep 76 - Sep 77 on Phase 1.

FOR THE

(6)

MODULAR SYSTEM CONTROL DEVELOPMENT MODEL (MSCDM). Sections 1-7.

DISTRIBUTION STATEMENT A

Approved for public release; Distribution Unlimited

for

THE DEFENSE COMMUNICATIONS AGENCY
WASHINGTON, D.C. 20305

(15) DCA100-76-C-0083

Burroughs Corporation

Federal and Special Systems Group

Paoli, Pa. 19301

070 040

78 12 11 178

CONTENTS

CONTENTS (Cont.)

CONTENTS (Cont.)

CONTENTS (Cont.)

CONTENTS (Cont.)

List of Figures

List of Figures

List of Tables

1. INTRODUCTION

This publication is the Phase I final report for the Modular System Control Development Model (MSCDM). It includes proposed designs, conclusions, recommendations, supporting analyses and rationale for the Phase II implementation of a Feasibility Development Model (FDM). This report is prepared by the Burroughs Corporation and is submitted in accordance with the requirements of Contract DCA100-76-C-0083.

1.1 Background

The following is based on information in the Statement of Work for Contract DCA100-76-C-0083.

Since 1974, the Defense Communications Engineering Center (DCEC) has been investigating various distributed computer architectures for various communication applications including digital speech processing, switching and system control. Distributed architectures that are designed from a functional decomposition point of view seem particularly interesting with respect to modularity, reliability and cost. Various efforts such as the Bolt Beranek and Newman's PLURIBUS and the Carnegie-Mellon University C.mmp advanced computer concepts have shown the advantages of a distributed architecture with respect to modularity, reliability and cost. For example, fail-soft behavior is possible, using these concepts, that will permit necessary functions to continuously operate even if some of the components fail.

The advent of Large Scale Integration technology and microprocessors in particular has now made it advantageous, cost wise, reliability wise, and maintainability wise, to design distributed computing systems. Microprocessors exist that can be used to replace wired

logic as well as sophisticated computing systems. In fact there are microprocessor systems on the market that are capable of replacing sophisticated minicomputer systems. Currently one of the main uses of microprocessors in the area of automatic control is the design of controllers. It now seems particularly advantageous for DCA to investigate the use of microprocessors in distributed architecture concepts which incorporate the functions of System Control as it pertains to the DCS. The DCS is a general-purpose system composed of leased and Government-owned transmission media, relay stations, and switching centers. It embraces all of the long-haul point-to-point DCS assets of the three Military Departments. The DCS encompasses a wide range of services, including command and control, intelligence, and early warning, as well as administrative and logistical communications. The major networks within the present DCS provide voice, secure voice, and secure record communications service. Each of these networks is characterized by a degree of automatic switching, a military precedence system, worldwide trunking, and service to a large community of defense and other U.S. Government users.

The control and management of a large communication system is a complex task. It includes the continual monitoring and assessment of system performance, the formulation and implementation of control actions in response to system performance degradation, the detection, isolation and restoral of faults and the generation of analyses, reports and displays in support of the system planning and engineering process. In order to carry out these functions, system controllers require ADP equipments which are geographically distributed and in constant interaction and communication with each other.

In recent years there has been considerable interest in more automated techniques for systems control. The Assistant Secretary of Defense for Telecommunications (now known as DTACCS) stated in guidance for submission of the FY 75-79 program objective memorandum that the DCA effort in the area of automatic system control and technical control should be expanded. The Defense Communications System (DCS) must be a highly survivable entity in order to insure that its vital mission is carried out. In order to enhance system survivability, it is highly desirable to decentralize the real-time monitoring and control process as much as possible. Therefore, if the system is fragmented the remaining system control elements will be able to effectively operate their fragmented portions of the DCS.

The DCS SYSCON Hierarchy is shown in Figure 1-1. MSCDM is primarily concerned with defining modules for the lower three levels (Sector, Nodal, Station). The goal is to define a set of modules that can be applied to the lowest level, and enhanced with additional modules to be applied to the higher levels.

The MSCDM functions must be specified as a set of functional modules. The requirements of these functions at the various SYSCON levels are determined. In Chapter 2 of this report the Level V Station Control is referred to as a Measurement Acquisition Station (MAS), the Level IV Nodal Control is referenced to as a Nodal Control Station (NCS), and the Level III Sector Control is referred to as a Sector Control Station (SCS). A single function

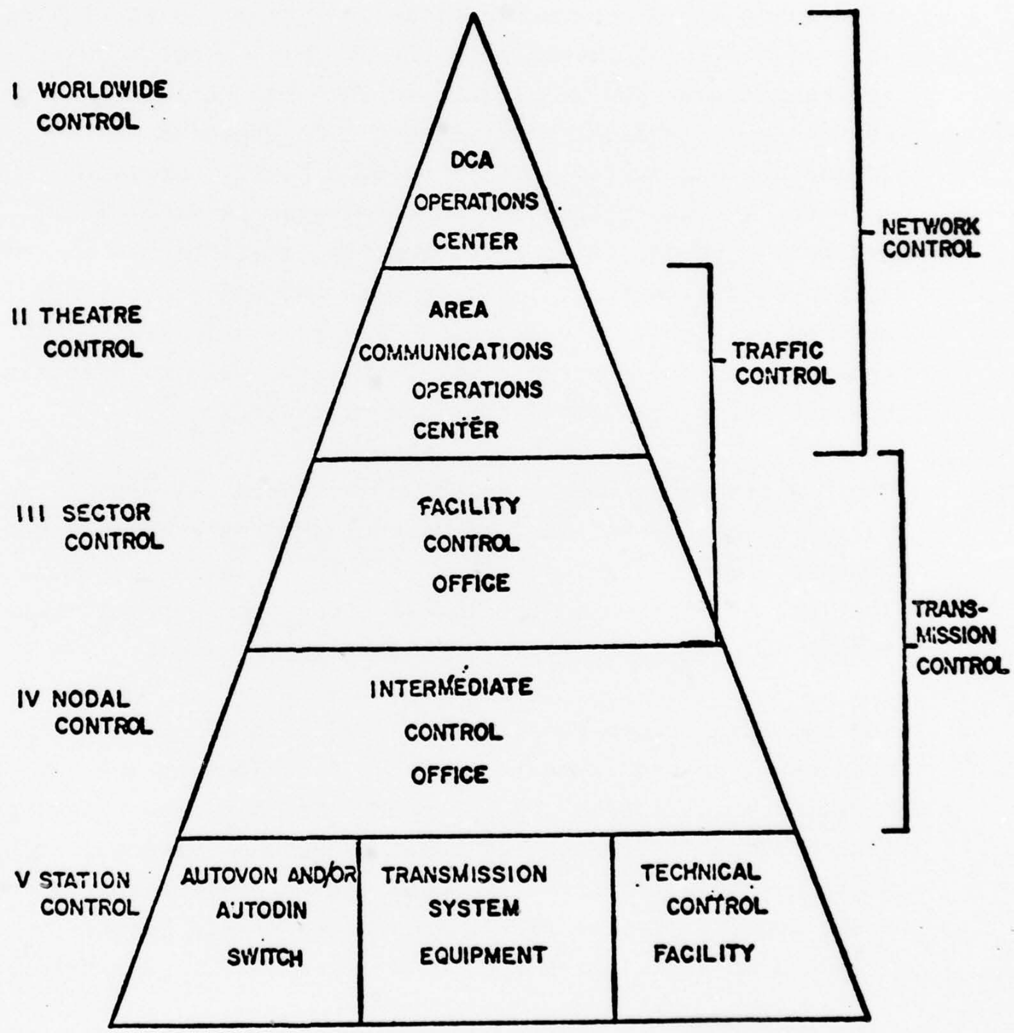


Figure 1-1
DCS SYSCON HIERARCHY

will usually correspond closely to a single software module, and therefore it is convenient to define the functional modularity (usually) as identical to the software modularity. These software modules are then mapped onto a set of hardware modules. For some functons, there is dedicated hardware which is therefore also a part of the functional modules, and in some cases this dedicated hardware will include the processing capability on which the software part of the functional module executes. Thus, the relation of hardware module to functional module can be one-to-one for some functions. Other functions, and other software modules, may well be multi-programmed onto a shared computational resource, whose hardware modularity is totally unrelated to the software modularity.

A function module has the following features:

- It is a construction module. Its interfaces to the rest of the system are so simple that it can be plugged in or out without disrupting the function of that part of the system not dependent on it. "Glue" in the form of wiring changes, program modifications, data base updates, and modifications of existing procedures, is not needed to add a module. The module includes all necessary features to make it work. There may be dependencies; module X may be a prerequisite for module Y, in that Y cannot be plugged in unless X is already there.

- It corresponds to a requirement. The set of modules at a site corresponds to the actual equipment that is being monitored at that site.

- The set of modules is complete. That is, any systems controller, at any level of the hierarchy, can be constructed by selecting the appropriate subset of the whole set. This completeness is conceptual, of course. We actually need only to construct the modules needed for a given station in order to construct that station.

Reasons for adopting such an approach to modularity include:

- Ease of construction of a new station

- Ease of repair

- System reliability through redundancy. Module independence leads to graceful degradation when the modules represent functionally dedicated hardware. Loss of one function does not cause the loss of unrelated functions.

- Diversity of station designs that can be assembled from standard parts

- Effects of program changes are confined generally to within the module involved. One change does not beget another.

- Maintainability through remove-and-replace procedures

- Scheduled growth by adding modules as needed

- Postponed obsolescence since individual modules can be updated as required, and new module types can be defined as the system grows.

Once the functional modules are defined the design of the FDM reduces to basically two system design level decisions. The first decision is concerned with which microprocessor should be selected as the basic hardware building block. The second decision is concerned with connecting the microprocessors to provide an interprocessor communication network. In this report, Chapter 4 is concerned with the first decision, and Chapters 5 and 6 are concerned with the second decision.

1.2 Purpose

As stated in the SOW this contract will accomplish the necessary developmental research, analysis, design and evaluation which will:

a. Provide design specifications of modular building blocks for future DCS system control distributed concepts that make maximum effective utilization of the microprocessor technology. Based on an analysis of these concepts, candidate architectures for these building blocks will be recommended that are of particular value in implementing the System Control functions.

b. Based upon the candidate architecture building blocks, provide a tradeoff analysis of software and hardware modularity concepts by characterizing their potential advantages and disadvantages to System Control functions with respect to life cycle support, performance, cost, operational capabilities, maintainability and reliability, and identify the best family of building blocks which can collectively perform the distributed System Control functions for the DCS.

c. Select from this family an optimum combination of modular building blocks which can perform the various functions required of system control, and size the capability of this combination of building blocks so that the combination can jointly and simultaneously perform each of the functions set forth under requirements.

d. Interface the combination of modular building blocks, delivered as a Feasibility Development Model, into the Exploratory System Control Model (ESM) so that follow-on experiments can be performed to test its effectiveness in accomplishing System Control functions and in its ease of modifying and maintaining modularly designed hardware and software systems.

The MSCDM is designed to implement the following functions:

1. Effectuation of systems control. This includes the acceptance of messages from an operator, interpretation of those messages in terms that are meaningful to the hardware, and the emission of messages that effect various results such as taking channels off-line, redistributing traffic, scheduling maintenance, etc.

2. Computations necessary for systems control. This includes trending of hardware states, traffic analysis, etc.

3. The control and scheduling of the performance monitoring and assessment function and the acceptance of the resulting data. The systems controller may include the interface to the monitoring hardware, if that hardware is found on the same site as the systems controller.

1.3 Selection Process

The microprocessor selection process is described in Chapter 4 of the report. From all possible microprocessors a small group (14) of candidates was selected. This group contained the more popular microprocessors (e.g., INTEL 8080), and did not contain micro processors that obviously could not become candidates (e.g., 4-bit microprocessors). This group was then reduced to two acceptable candidates (TMS 9900, LSI-11) by eliminating the microprocessors as candidates if they failed to satisfy one or more features of a

desirable features list generated from the MSCDM requirements. The two acceptable candidates were then directly compared including a benchmark test that was run on the two candidate microcomputers, their compatible minicomputers, and the Burroughs B776. The recommended microprocessor was the TMS 9900 based primarily upon its lower cost and higher speed as compared to the LSI-11. It was estimated that an additional microprocessor would be required at an additional system cost if the LSI-11 was used in order to satisfy the FDM requirements stated in the SOW.

The distributed architecture for microprocessor communication selection process is described in Chapters 5 and 6 of the report. From all possible architectures a small group (6) of candidates was selected based upon the assumption that the cost of connecting microprocessors should not be significantly larger than the cost of the microprocessor modules. The six candidates investigated were BBN/PLURIBUS, CM*, SUNY, MINERVA, ETHERNET and LOOP. The candidates were compared with respect to the sensitivity analysis criteria (i.e., modular construction, cost-modularity, place-modularity, throughput, simplicity of interface, reliability, cost, software maintenance, adaptability/flexibility, and low implementation risk). Based upon the analysis with respect to the MSCDM application, the candidates selected were the single bit serial busses without bus arbiters; i.e., the ETHERNET and LOOP. These candidates were found to be less expensive, easier to implement, easier to interface, and more modular.

The ETHERNET and LOOP, with respect to a specific FDM design, were then compared in detail (Chapter 6) including a life-cycle costing analysis and simulaton study. The LOOP was found to be significantly superior to the ETHERNET, and thus become the recommended architecture. Of the four proposed FDM system designs (LOOP-TMS 9900, LOOP-LSI-11, ETHER-TMS9900, ETHER-LSI-11), the LOOP-TMS9900 was

found to provide the best performance at the lowest cost. Based upon the above analysis and experience obtained in building other LOOP networks connecting microprocessors (e.g., ADO, ESM, ESMD, ADVISOR), the Burroughs Corporation is confident that a powerful FDM-simulation facility can be implemented in a cost-effective and timely manner in Phase II.

1.4 Report Organization

A description of the various chapters of this final report are presented below:

Chapter 1 provides background information and outlines the selection process used in determining the recommendation of the TMS 9900 microprocessors connected by a LOOP communication network.

Chapter 2 provides a functional analysis and decomposition of the MSCDM functons. The SYSCON DCS levels are mapped to "column" functions and "row" functions. Column functions are suitable for mapping onto microprocessors, and they are made up of "row" functions which are suitable for mapping onto software modules (e.g., FORTRAN subroutines).

Chapter 3 provides module specification and definition. A detailed description of the flow of informaton and control in the function modules suitable for hardware/software mapping is presented.

Chapter 4 provides a microprocessor study and analysis. A group of candidates is reduced to two acceptable microprocessors (TMS 9900, LSI-11) determined from a list of desirable features with respect to the MSCDM application. The two candidates are compared in detail including a benchmark test. The recommended microprocessor is the TMS 9900 based upon lower cost and higher speed.

Chapter 5 provides an architecture analysis. Six candidates are compared with respect to the sensitivity analysis criteria. The single bit serial bus architectures (ETHERNET, LOOP) are selected as final candidates.

Chapter 6 provides a candidate architecture analysis and design. Four designs are compared: LOOP-TMS9900, LOOP-LSI-11, ETHERNET-TMS9900, ETHERNET-LSI-11. An ESM interfacing approach is described. The architectures are compared in detail including a life-cycle costing analysis and simulation analysis. The recommended architecture is the LOOP-TMS9900.

Chapter 7 provides a description of the user language that will be utilized to operate the FDM.

Appendix A contains the SYSCON Data Base Study performed by UTEK Systems. The study details the findings based on UTEK's analysis of the reporting procedures and support networking used for monitoring and management of communications by the Defense Communications System (DCS). The Appendix contains the report submitted by UTEK and eight sub-appendices.

Appendix B provides a tutorial on loop communications systems. The loop architecture is the recommended architecture for the MSCDM application. Example loop systems built by Burroughs (ADO, ESM, ESMD) are described in detail. One loop system currently being built that is not described in Appendix B is the Advanced Information System Organization (ADVISOR) loop. This loop uses hardware similar to the ESMD loop (e.g., BDS microprocessors). This program is a join effort between Burroughs Advanced Development Organization, Paoli, Pennsylvania, and Burroughs government systems marketing office located in McLean, Virginia, to develop a pilot system of an "Office of the Future". The objective is the development of various

techniques which combine data processing capabilities with that of word/text processing and include digital voice and visual data on the same network. The McLean office and the Paoli office are linked by leased line giving either office access to data/program files and allowing "electronic" mail delivery between the two locations.

Appendix C lists the LOOP and ETHERNET simulation outputs generated by the Burroughs Operational Systems Simulator (BOSS) run on a B6700.

Appendix D provides a Glossary of Acronyms.

2. FUNCTIONAL ANALYSIS AND DECOMPOSITION

2.1 Functional Analysis

2.1.1 Introduction

Sections 2.1 and 2.2 of this report discuss tasks 1 and 2 of Phase I of the Modular System Control Development Model Program: Analysis of Communication System Control Functions and Decomposition of These Functions. The approach taken in these discussions views the functional requirements of the MAS, NCS, SCS levels of the DCS and System Control network as "living" in a plane. Along one edge of the plane are functional areas which are dependent upon the equipments and reporting facilities located at a site (e.g., switch data facilities, voice channel facilities, wide-band digital facilities). These functional areas are generically referred to as 'column' functions. Along the orthogonal edge of the function plane are the 'row' functions. Row functions represent types of operations which must be accomplished in order to implement column functions. The utility of this approach is seen in the similarities of structure and function of what are called the column functions. This report does not attempt to be exhaustive in the coverage of column functions; rather, its purpose is to demonstrate feasibility of the implemetation of column functions by means of a distributed microprocessor system. It is intended that the intersections of row and column functions will characterize the modules to be discussed in context of Task 3.

Section 2.1, Functional Analysis, discusses the row functions, primarily with respect to sizing. The sections represent the row functions as follows:

1. DC - Data Collection
2. DR/DA - Data Reduction/Data Assessment
3. DB - Data Base organization

4. R - Reporting
5. CCI - Command and Control Interpreter
6. SC - Scheduling of functions
7. CI - Communications Interface

Section 2.2, Functional Decomposition, addresses the column functions in terms of their decomposition into row functions as follows:

1. VSQC - Voice Service Quality Control
2. DSQC - Digital Service Quality Control
3. BBSA - Baseband Signal Analysis
4. WBSA - Wideband Signal Analysis
5. SDCA - Switch Data Collection/Analysis
6. OCRI - Operator Control and Report Interface
7. FIAC - Fault Isolation and Analysis Coordination
8. SSCI - Station to Station Communications Interface
9. DBMS - Data Base Management Service

The discussion in Parts 1 and 2 also assumes the following configuration of components at the MAS level:

1. Two switches - any combination of AUTOVON and AUTODIN.

2. Three links - radio type consisting of appropriate transmitters, receivers, and multiplexors (either digital or analog).

3. 1000 channel appearances.

2.1.2 DC - Data Collection

This function is responsible for the collection of data necessary to the column functions of which it is a part. The DC components discussed in this section are:

1. DC (BBSA)
2. DC (WBSA)
3. DC (VSQC)
4. DC (DSQC)
5. DC (SDCA)

2.1.2.1 DC (BBSA) - Baseband Signal Analysis Data Collection

This function provides the interface to the transmitters, receivers, and multiplex equipments for analog links. The data collected by this function consists of both 'hard' alarms and metered data accumulated either by direct digital interface or through analog/digital interfacing to the system. The following list is representative of the alarms and metering that must be taken into account in the Feasibility Development Model design:

1. Transmitter

   a. Alarms

      fuses/breakers
      transmitter frequency
      transmitter power

   b. Metering

      percent modulation
      transmitter deviation
      relative transmitter power

2. Receiver

   a. Alarms

      fuses/breakers
      phase lock loop failure

   b. Metering

      AGC voltages
      receiver IF output

3. Multiplexor/Demultiplexor

   a. Alarms

      fuses/breakers
      DeMux frame sync loss
      Mux output data loss
      DeMux input data loss
      group carrier alarms

   b. Metering
      multiplex pilot levels
      baseband levels
      slotted noise levels
      group level
      channel level

The alarm status indicators must be sensed such that the latency of response from an alarm condition being raised to system notification should be on the order of 30 seconds. With respect to metered data, out-of-range conditions as determined by thresholding must be detected within 15 minutes. These times are based on the MITRE/ESD ATEC

System Description of December 1, 1976. Based on the general assumptions, the total number of metered signals which must be accounted for with respect to scanning of the equipments associated with BBSA are as follows:

	3 transmitters * 3 signals	=	9
+	3 receivers * 2 signals	=	6
+	8 baseband levels	=	8
+	5 group pilot levels * 8 basebands	=	40
	Total		63 signals

Thus in order to meet the requirements for response time in detection of out-of-range conditions, the Feasibility Development Model must be able to process a measurement every 14 - 15 seconds.

2.1.2.2 DC (WBSA) - Wideband Signal Analysis Data Collection

This function provides the necessary interface to the transmitters, receivers, and multiplex equipments for wideband digital links. The data collected by this component consists of both 'hard' alarms and metered data accumulated by direct digital interface to the system. The following list of alarms and metered data is based in part on those suggested in Advanced Monitoring Techniques for Digital Communication System, Georgia Tech Research Report E21-655, 1976:

1. Transmitter

   a. Alarms

      fuses/breakers
      transmitter frequency
      transmitter power

b. Metering
percent modulation
transmitter deviation
relative transmitter power

2. Receiver

a. Alarms

fuses/breakers
phase lock loop failure

b. Metering

AGC voltages
receiver IF output

3. Multiplexor/Demultiplexor

a. Alarms

fuses/breakers
out-of-band noise
format violation greater than 7
out-of-frame indication

b. Metering

pseudo-error

Notice that signal power, which is a suggested measurement of the Georgia Tech report, is included in receiver metering. The metered signal profile will be effectively the same as for DC (BBSA) except that pseudo error will be measured for baseband levels and group pilot levels instead of for signal levels. The same response constraints as mentioned in DC (BBSA) thus require a scan rate of one measurement every 14 - 15 seconds as with DC (BBSA).

2.1.2.3 DC (VSQC) - Voice Service Quality Control Data Collection

The source data necessary to assess channel/signal quality is the signal level at an appearance, in the case of a voice channel. The nominal bandwidth of such a channel is taken as 4 KHz for the AUTOVON network. Assuming an initial stage of low pass filtering with a cut-off of 6.4 KHz, the acquisition rate of the A/D interface to the appearance must be 12.8K samples/second. The SOW indicates that the Feasibility Development Model must be capable of analyzing 1000 appearance/hour; this represents a sample rate of 5.7K samples/second if one assumes that 20,480 samples are required to analyze one channel, as in AN-GYM-13(v). Thus the acquisition rate of 12.8 KHz dominates and suggests that a single A/D interface might suffice. Associated with the A/D interface, it is assumed that some form of multiplexor is attached as an I/O device accepting a command packet and indicating which appearance to connect the A/D to, as well as bridging impedance, and gain setting informatin. The A/D resolution will be 13 bits in signed-magnitude representation.

2.1.2.4 DC (DSQC) - Digital Service Quality Control Data Collection

In the case of digital signal quality the following parameters are assumed to be directly measurable for each appearance:

1. Pseudo error rate
2. Bit error rate
3. Block error rate
4. Distortion Measurement (e.g., bias, Fortuitous, peak total distortion)

In order to monitor 1000 appearances/hour of digital channels, the above three parameters must be collected and analyzed at the rate of one sample every 1.2 seconds. Each parameter is assumed to require no more than 16 bits of representation in the system. A multiplexor for selecting appearances will be assumed as above.

2.2.2.5 DC (SDCA) - Switch Data Collection/Analysis Data Collection

Switch traffic data collection occurs over a 2400-baud synchronous circuit on a per-switch basis. The Feasibility Development Model will have the ability to transmit requests over this circuit to gain access to the routing tables of the switch. Traffic data from the switch is assumed to consist of 1K bit messages every 5 seconds for each of the following parameters:

1. Number of incoming and outgoing transactions by precedence
2. Number of blocked transactions by precedence
3. Transaction queue depth
4. Number of preempted transactions
5. Trunk group occupancy
6. Trunk group overflow
7. Message delay by precedence
8. Maximum message age by precedence
9. Number of overflow messages
10. Common switch equipment usage profile:
    a. senders
    b. markers
    c. receivers
    d. pooled crypto units
11. Call service time for:
    a. dial tone
    b. crypto unit

In the above, 'transaction' refers to calls or messages as appropriate to the kind of switch. The above assumptions imply that the maximum utilization of the switch traffic collection channel will be 94 percent, which represents a parameter group every .42 seconds. The figures must be considered an upper bound on parameter group arrival since some parameters will not require 1K bits, and not every parameter group is applicable to a given switch type.

2.1.3 DR/DA - DATA REDUCTION/DATA ASSESSMENT

This section discusses the processing requirements, exclusive of data base organization, with respect to the collected data. Data reduction is applied to low order metered data, i.e., digitized signal level, in order to derive higher order parameters which directly reflect performance characteristics of a monitored point. Performance assessment is applied to high order parameters in order to detect degrading or degraded conditions with respect to a monitor point. The general information flow of this functional area is depicted in Figure 2-1. The function of alarm handling is discussed later in conjunction with the discussion of fault isolation, reporting, and operator interface. The general requirements of degraded and degrading conditions are discussed and then followed by a discusison of the derivation of parameters from the digitized signal level of in-service assessment.

2.1.3.1 DA (mf)*

Degraded conditions are detected via a comparison of the current value of a parameter with a threshold value which has been determined a priori to be significant of a condition of failure for the monitor point associated with the parameter. Such thresholding is intended to be indicative of an instantaneous change in state of monitor point performance. A single threshold may be considered to reduce a parameter to a two state variable: e.g., OK/not OK. In practice, however, two windows (upper and lower thresholds) are often associated with a parameter to reduce it to a three state variable, e.g., GREEN, AMBER, RED (see figure 2-2). The RED region is indicative of a total degradation performance. The AMBER region is chosen to represent an operating region of the parameter which is marginal in performance but still functional. AMBER is used to warn the controlling elements of the system of an impending failure so that isolation and corrective actions can be effected in anticipation of failure. One aspect

*multifunction, where mf can be VSQC, DSQC, BBSA, WBSA.

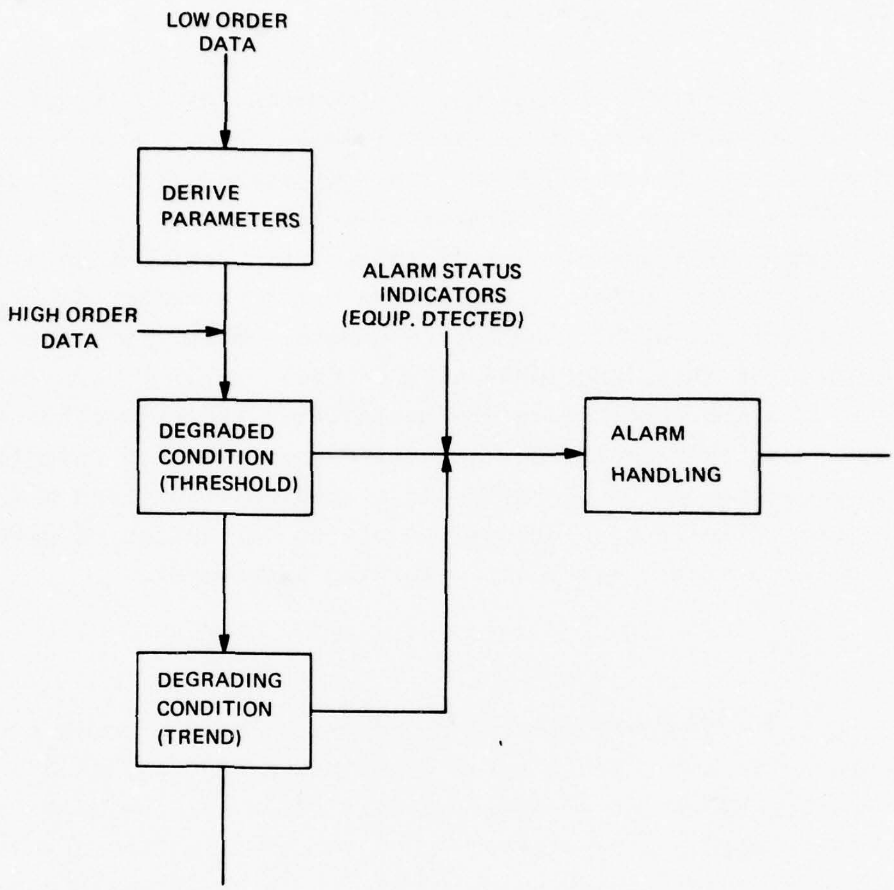


Figure 2-1
Performance Assessment Information Flow

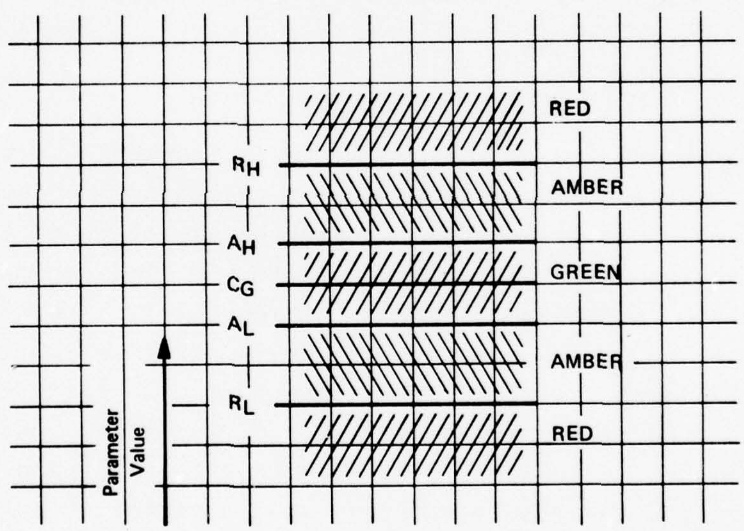


Figure 2-2
Parameter Thresholds

of this anticipation is the requirement that the alarm state of the parameter be verified to determine whether the condition is stable or transient. Thus a possible strategy would be to report the first transition beyond a threshold as an anticipated new state of the parameter, and declare the new state of the parameter after verification. The new state would be the one acted upon. Verification would consist of remeasuring the parameter immediately instead of waiting for the normal scan sequence. It is appropriate to note, however, that the particular strategy employed is in general dependent upon the failure characteristics of the particular parameter. For example, a parameter such as relative transmitter power would be amenable to a simple strategy such as outlined above; however, receiver AGC voltage would require a more sophisticated strategy to account for anomalies, such as fading, which are not directly indicative of degradations of equipments.

A further consideration is the necessity to 'dampen' the reporting of state changes which oscillate about a threshold boundary. In such a case the worst state should be considered the state of the parameter as long as the parameter shows an oscillation. The general flow of the thresholding procedure is shown in Figure 2-3. The computation sizing of the above flow consists of some small fixed part (1) + (3) and a variable part (2) , which is dependent upon failure characteristics. The fixed part consists of five compare operations and is easily seen to be insignificant in both time and space. The variable part for any likely strategy should certainly be no more than several hundred operations; hence, thresholding should not represent a significant implemetation constraint.

In general, trending of parameter time series is employed to detect or predict longer term degradations in performance, due typically to the decay or drift of components. For the purposes of this report, the CSC-FS4952-00320 report on 'Adaptive Trend Analysis' is used. In summary, the approach involves a polynomial model of the trend of the parameter, where the order of the polynomial is 0, 1, or 2. A digital filter is applied to accomplish prediction. For short term (on the order of an hour), the filter has an adaptive smoothing

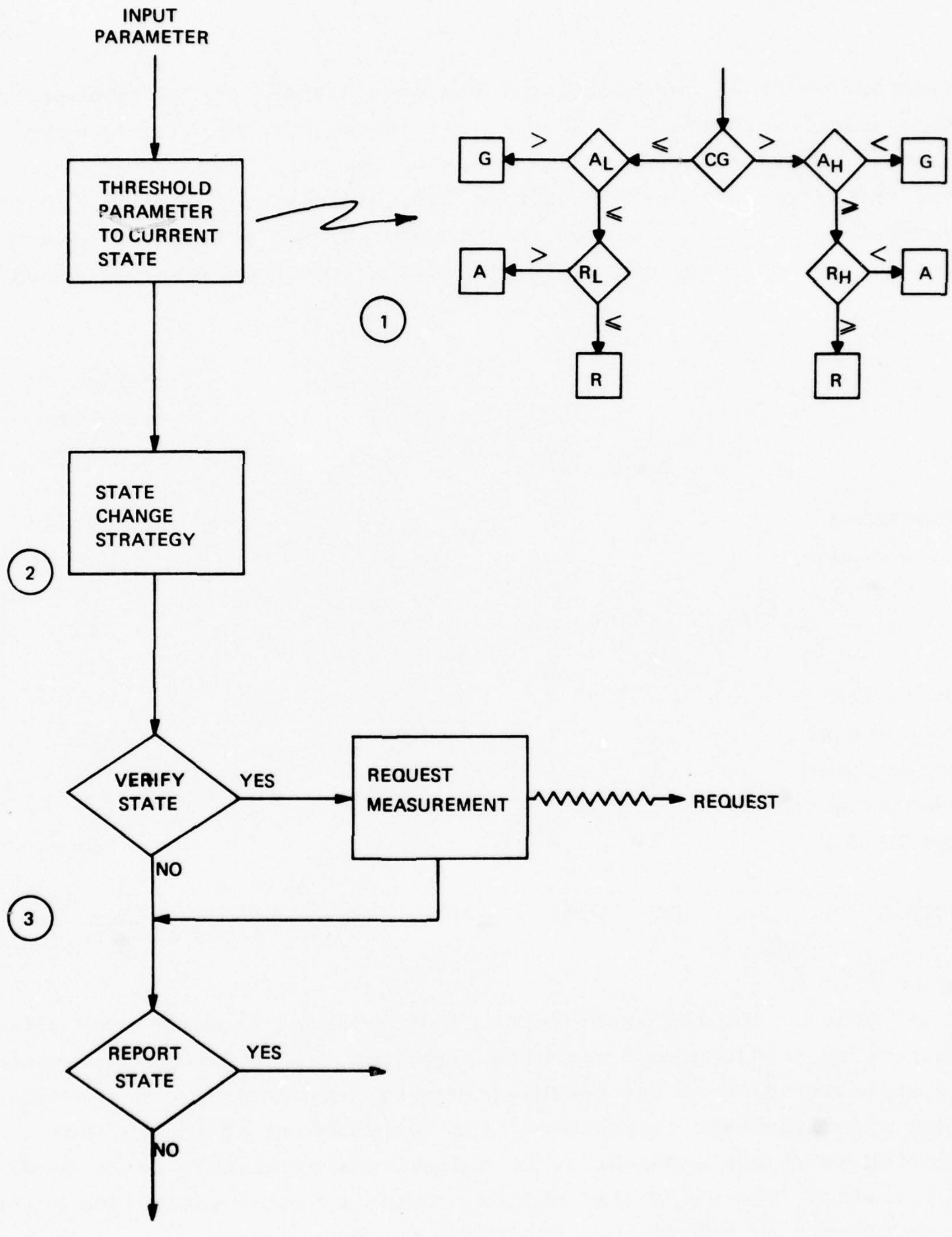


Figure 2-3
Thresholding Procedure Flow

constant which is dependent upon the past history of the time series. This adaptive approach is in contrast to the AN-GYM-13(v) approach of simply rejecting 'wild' fluctuations of the parameter. Further, the technique includes a mechanism for determining a best fit to the parameter in terms of one of the above mentioned polynomial orders. The following is an adaptation of Table 1 from the above mentioned report:

	Add	Sub	Mul	Div	Sto	HP21MX (max time In usec)
Smoothing	3	1	6		3	600
Prediction						
order = 0					1	4
1	1	2	4	1		483
2	12		19	1		1851
Meas. diff		9			5	533
Corr funct.	2	2	4	1		543
Noise power	1	1	3			295
Smoothing const.	1			1		132
Mean & S.D.	2	2	2			340
TOTALS	22	17	38	4	9	4295

This table is for the short term trending, hence is worst case with respect to the long term trending algorithm. This data is presented as representative of the overhead incurred in trending a parameter. This algorithm must be followed by a thresholding as in the last section to reduce a parameter to a three state variable as in AN-GYM-13(v). The timings from the HP21MX should provide a useful comparison point with respect to the various candidate microprocessors.

2.1.3.2 DR (VSQC) - Voice Service Quality Control Data Reduction

As mentioned earlier, the VF signal level parameter is considered a low order parameter. The parameters to be derived from VF signal level are:

1. Peak power (dBmO)
2. Average power (dBmO)
3. Peak to average ratio (dB)
4. 3 KHz flat noise (dBmO)
5. C-msg noise (dBmO)
6. Spectrum display (dBmO in 100 Hz increments)

The following discussion is based on the practices employed in AN-GYM-13(v), IQCS. The data is collected as 32 frames of 5 packets, each of 128 sample points. Each packet is thus 10 milliseconds in duration. The five packets/frame are collected contiguously and followed by 50 milliseconds of computation time. The samples are written as V(i,j,k), where 1 i 128, 1 j 5, 1 k 32. The peak power may be expressed as:

PEAK = MAX(ABS(v(i,j,k))**2) 1)

This represents a total of 20,480 ABS and MAX operations. It may be possible to reduce the number of operations by using fewer samples as discussed below. The average power estimation is expressed as:

AVG = SUM (V(i,j,k)**2) / 20,480 2)

This would require 20,480 MUL and ADD opera ions; however, the ABS in equation 1 may be employed to compute a 'relative' average power estimation as:

RAVG = (SUM(ABS(v(i,j,k))) / 20,480)**2 3)

This approach requires 20,480 ADD operations in addition to equation 1 to get an estimator of average power which is in many cases proportional to equation 2, at considerably less computational cost. Since RAVG will not weigh peaks in the signal as much as equation 2, the peak to average ratio will, in general, be larger than if computed using equation 2 for high crest signals. For low crest signals, the ratios should be more in agreement.

In AN-GYM-13(v), the spectrum display is computed as the average over 32 FFTS. Each FFT is computed using the V(i,5,k) packet of samples. The number of real multiply and add operations required per FFT may be computed using Bergland(68) as:

MULs = (7- 3.5)*128 + 6 = 454 per FFT

and

ADDs = (1.5*7 - 1.5)*128 + 2 = 1154 per FFT

For all 32 FFTs, the total number of operations is:

MULs = 14,528 and ADDs = 36,928 4)

Assuming a total sample size of 20,480 for one VF appearance analysis, we may summarize the two alternative approaches discussed thus far:

	1)+2)+4)	1)+3)+4)
ADD	57408	57408
MUL	35008	14528
ABS	20480	20480
MAX	20480	20480

The number of samples used to compute the peak and average power estimators may be reduced to 4096 by using only those samples which are used in the computation of the FFTs. The justification for this

statement is that for steady traffic types such as 'idle dead' and 'test tone' this number of samples does not significantly impair the error characteristics of the estimates, if at all, and for voice traffic over a 3.2 second sampling period, it is not clear that the additional 16,384 sample points would give any better estimates of the two power parameters owing to the stationary period being so short compared to the time scale over which this nonstationary process varies. Thus, two more alternatives to deriving the parameters can be given by considering 32 frames, each consisting of 1 packet of 128 points, with a 90 millisecond computation period between packets. Summarizing these additional alternatives:

	1)+2)+4)	1)+3)+4)
ADD	41024	21024
MUL	18624	14528
ABS	4096	4096
MAX	4096	4096

These four alternatives are provided to allow for trade-offs to be made with respect to various microprocessors and module architectures. The storage requirements depend on how much computation can be accomplished within a time frame. The best case would result in 128+40+2 = 170 cells; the worst case would require on the order of 20,480 cells. The particular strategy of course depends on the power of the particular microprocessor. This particular requirement represents the worst case computational requirements of the system from a number crunching point of view. If the system can handle this requirement, the other functions in this section should be easily accomplished.

2.1.4 DB - Data Base Organization

This section considers the necessary structures and information items which must be accessible in an on-line manner to the Feasibility Development Model. The amount of storage and the access time require-

ments will be estimated, insofar as possible, to be consonant with the current European DCS. In accordance with the approach to information storage requirements indicated in the MITRE ATEC system description of Dec. 1, 1976, the Feasibility Development Model simulation system storage should be large enough to hold the complete connectivity of the European theatre at the SCS level. At the NCS level, the connectivity of an entire sector must be kept. Other information storage requirements which must be accounted for are the MAS requirements for monitor point information such as trend status and associated parameters, alarm status and parameters via thresholding, etc. Miscellaneous storage for programs, report forms, and operator files of commands should be a small fraction of the requirements.

In order to estimate the request arrival rate, one needs to know the rate of parameter measurement or derivation, since in general each parameter will require a fetch and a store to the data base. The response time necessary for an access is dependent upon the module architecture and number of modules simultaneously requesting access. For example, if one has a very fast access time, say 500 milliseconds, this may lead to fewer IQCS type modules in the system for a given number of appearances per hour than would be necessary if there were a 5 second access time. Likewise, if the access time is short, one may need fewer data base modules than would be required otherwise in order to satisfy the arrival rate without creating lengthy queues of requests. Thus, this section can only indicate the arrival rate and say little about response time. The response time must be analyzed with respect to particular hardware alternatives.

The primary source of connectivity data base requirements is taken from DCA 310-65-1: Circuit and Trunk Files, Data Elements and Code Manual; June 1976.

2.1.4.1 MAS Requirements

At the MAS level, the storage requirements are determined by the parameters which are collected and the trending and alarm thresholding requirements for a parameter.

The analysis of the data base requirements at the MAS level is presented here. If we assume the CSC trending algorithm and a reasonable number of threshold alarm data items we can make the following estimate of space:

Short term trend	18 items
Long term trend	9
Threshold alarming	10
Total	37

Each item is represented in 16 bits or 2 bytes; hence, each parameter with the above assessment menu requires 74 bytes. Among the parameters that we would expect to trend, we have for VF channel assessment, C-msg noise and 3 KHz flat noise. This gives at least 148 bytes per channel appearance. Each appearance requires information as to the monitor point address, the CCSD as a key, gain, spectrum display and impedance items. We therefore estimate 256 bytes per channel appearance. For each link equipment measurement, for example group pilot level, we assume 256 bytes of storage. If in regard to the switch data, data base transfers are based on all the parameters mentioned in the previous section, this is a 1.3 K byte packet of data every 5 seconds. For each channel assessment we have to read a packet and store the updated packet; likewise, for the link measurements. The traffic data is stored as a history file so that for each collection period we must store a packet.

The activity rates are computed based on the following:

1000 appearances/hour * 2 accesses	= 2000 accesses/hour
63 link measurements/.25 hour *2 accesses	= 504 accesses/hour
1 switch packet/5 secs * 2 switches * 1 access	= 1440 accesses/hour
Total	3944 accesses/hour

In order to compute the transfer bandwidth in K bytes/second:

2000 * 256	= 512 K	bytes/hour
504 * 256	= 129 K	bytes/hour
1440 * 1.3 K	= 1872 K	bytes/hour
Total	= 2513 K	bytes/hour
or approx.	.7 K	bytes/second

The number of access requests is on the order of 1 every .9 second under the above assumptions.

The total storage represented by the above assumptions depends on the time span of the switch history file. The major activity regarding switch data would be to transfer it up the hierarchy to be analyzed so that the storage requirements for switch data can be viewed as backup. Something on the order of 2 hours of storage does not seem unreasonable. The following is a summary of the storage requirements represented thus far at the MAS level:

1000 appearances * 256 bytes	= 256 K bytes
63 link measurements * 256 bytes	= 16 K bytes
1872 K bytes/hour * 2 hours	=3744 K bytes
Total	=4016 K bytes

In addition to these requirements we must store the necessary report forms to be used by the tech controllers, the programs, and schedule tables. A maximum estimate of 4.5 M bytes of storage to model the MAS level seems reasonable. Since it is desirable for minidisk to be used for the Phase II FDM equipment, a subset of the above requirements will be implemented.

2.1.4.2 SCS, NCS Requiremetns

In order to estimate the data base storage requirements at the Sector and Nodal levels, the following estimates for DCS Europe are used:

1.	Circuits	48,000
2.	Trunks	4,000
3.	Segments	5 per circuit or trunk

Following DCAC 310-65-1, the estimates for the circuit file are based on the following record sizes and numbers of records on a per circuit basis:

1.	Basic circuit header-1	85 bytes
2.	Basic circuit header-2	33
3.	User terminal header	66
4.	5 circuit segment records	5*61
5.	Diverse and avoidance routing	55
6.	Contingency plan circuit header	53
	Total	597

Similar estimates for the trunk file are:

1.	Trunk header record-1	79
2.	Trunk header record-2	36
3.	5 trunk trailer records	5*56
4.	12 channel assignment records	12*62
	Total	1139

Since each sector is required to maintain the connectivity data base for the areas of which it is the member, the sector level storage requirements may be obtained as approximately

600 bytes/circuit * 48,000 circuits	= 28.8 Mbytes
1200 bytes/trunk * 4,000 trunks	= 4.8 Mbytes
Total	= 33.6 Mbytes

If a sector is assumed to comprise approximately one fifth of an area, then the nodal level data base storage requirements may be estimated at 6.7 Mbytes; since each nodal station must maintain the connectivity data base for the sector of which it is a member.

The request arrival rate for the connectivity data base should be in direct proportion to the rate and severity of fault detection, which is assumed to be no greater than the estimates for the MAS level data base of 1 request every .9 seconds.

2.1.5 R - Reporting

The reporting, R, component of a function is responsible for constructing and forwarding the results of the function execution to a predefined receiver, determined in general by the type of result.

For the VSQC, DSQC, BBSA, WBSA functions, the R component generates two basic types of report: a) event and b) requested. Event reports are the result of a threshold or trend violation having been detected within the function. These reports are distributed to FIAC and OCRI. Requested reports may be the result of either FIAC or OCRI commands to the function. These reports are forwarded to the requestor only.

In the case of R(SDCA), all reports are treated as requested reports. Normal switch traffic reports are addressed solely to the next higher level SDCA. OCRI requested status reports are of course to be forwarded to the requesting OCRI.

2.1.5.1 R(OCRI)

There are three primary reporting requirements which must be met:

1. Status reporting
2. Facility and link reporting
3. Traffic data reporting

The source for status reporting requirements is DCAC 310-55-1: Status Reporting for the Defense Communications System. Status reports are to be generated in response to outages of services or equipments under the responsibility of the station. The circular specifies two types of reports:

1. Nonformatted report (NR). A narrative report of DCS status required as soon as feasible after a reportable event occurs.

2. Formatted report (FR). A formatted report containing status information on previously reported items and other DCS status information.

The system requirements for this type of report are twofold:

1. Assist operator in the preparation of the report by providing formed displays.

2. Forwarding of these reports to the next responsible level in the hierarchy.

Since it is assumed that the Feasibility Development Model will be collecting switch traffic data, the [illegible]ONDATA and DINDATA recoverable subjects will be handled as a separate reporting function, rather than as a formatted narrative report.

The source for facility and link reporting requirements is DCAC 300-85-1: Reporting of DCS Facility and Link Data. The purpose of these reports is to furnish DCAOC with the information necessary to update the Facility/Link Data Base. This information represents the profiles of physical elements of the DCS; hence, reports are required for each change in status of the profile of the responsible site. The system requirements for this type of reporting are the same as for the status reports.

Traffic data reporting consists of transferring up the hierarchy the data collected at the MAS level from the switch, as outlined in the section on data collection. It is assumed that the NCS and SCS levels of operation will reduce the traffic data for further reporting to higher levels with provisions for storing history information at each level. This reporting function is expected to be transparent to the controller at the MAS level. The MAS controller would be notified of any failures in the reporting mechanism of traffic data that represented faults in the level-to-level communications facilities.

2.1.6 CCI - Command and Control Interpreter

The purposes of this function are to receive commands from other functions and to effect the commands after interpretation by interaction with other components as necessary within a column function. In the case of the FIAC, CCI also issues commands to other functions (see Section 2.2.8 - FIAC).

2.1.6.1 CCI(mf)*

The Status Report request returns current monitor point location, and alarm status.

The Monitor and Display request causes a specified monitor point to be assessed and results forwarded by R(mf) to OCRI for display. An optional repeat count specifies the number of times to assess the given monitor point. In effect, the command is placed at the head of the schedule queue.

---

*mf can be VSQC, DSQC, BBSA, WBSA.

The Monitor Sequence requests SC(mf) to set up a queue of monitor commands based on a list of monitor points contained in the request.

The Display request causes R(mf) to forward current results and status to OCRI for display.

The Idle request results in mf holding in CCI(mf) until further commands arrive. In the case that R(mf) issues an event notice to FIAC, mf is forced to this state to await acknowledgement from FIAC.

The Clear request causes the schedule queue to be flushed, and then enters the idle state.

2.1.6.2 CCI(SDCA)

The Restart request causes the SDCA function to begin accepting switch traffic data for processing.

The Idle request results in SDCA holding in CCI(SDCA) until a restart command arrives. Switch traffic data which arrives while in idle is lost.

The Status request causes R(SDCA) to forward to OCRI the current state of SDCA, i.e., idle or active.

2.1.6.3 CCI(OCRI)

This function interprets the controller input strings and results in the forwarding of the appropriate commands to the necessary functions. The details of the input strings are relegated to the report on Task 5: Operator Language Definition. The results of requests to other functions for status or display information are treated as commands to OCRI for driving the display device of the OCRI. The

controller reporting function is handled by a command to DBR located with OCRI for the appropriate form. The result of the DBR action is again treated as a command to OCRI for driving the display. The controller input is then handled by R(OCRI).

2.1.6.4 CCI(FIAC)

This component is discussed in Section 2.2.8 - FIAC

2.1.7 SC - Scheduling

It is necessary to provide several capabilities that may be regarded as scheduling in the classical sense. It is possible that the function of scheduling may be distributed in the sense that no single processing element in the system can be said to be responsible for carrying out the function of scheduling.

There are two primary requirements in this area:

1. Sequences of measurements which are made 'automatically'
2. Measurements made on command, either by the controller or from a level up in the control hierarchy.

It must be possible to specify that a group of super-group of channel appearances be scanned in some order either once or repetitively. For the purposes of alarm verification and fault isolation it must be possible to request either single or repetitive measurements of a single monitor point. Further fault isolation may require the coordination of several measurement capabilities at one time. The mechanism for accomplishing these various objectives is termed scheduling.

A way of thinking about the scheduling in a distributed fashion is to consider each monitor (VSQC, BBSA, etc.) as possessing a queue of monitor commands. These commands may take the form of (monitor-

point, how-many). This queue may be built by the monitor itself based on some function stored in the data base or the queue may be built by some external agent, for example related to operator interface. In order to handle the various forms of scheduling requirements, the agent that makes entries in the queue must have the ability to insert-at-head, insert-at-tail, or flush.

2.1.8 CI - Communication Interface

This is the major component of the SSCI function (see Section 2.2.9). This component performs interstation routing on functional addresses contained in message headers. The routing is applied only to inter-station messages. Messages within a station are simply addressed to the function by name, for example:

SDCA#2//information//

Interstation messages are addressed by station name and function. For example:

NCS FIAC//information//

could be issued by some MAS level station. Another example might be:

NCS#3 FIAC//information//

issued by an SCS level station. Since the topology indicated by the MITRE/ESD Atec System Description allows only one NCS - MAS connection, no NCS# is allowed; however, in going from SCS to NCS or NCS to MAS there will be multiple stations; an appropriate indicator of which station is required.

Another requirement of the CI is handling controller to controller communications, for example between MAS stations. In this case messages are addressed:

MAS#nOCRI//information//

and CI must route hierarchically as necessary depending upon the location of MAS#n. In this case 'n' must be network unique. So

if 'n' is under the control of the same NCS as that controlling the sender, the CI at that NCS can directly route the message. If the NCS controlling the sender MAS does not recognize 'n' then the message must be routed to the controlling SCS. The SCS CI then either routes to a directly controlled NCS or to the controlling SCS. The 'n' can be constructed very simply by concatenating SCS#NCS#MAS#.

In this case MAS# is unique within a particular NCS, NCS# is unique with a particular SCS, and SCS# is the only network wide unique address. This type of address is essentially self routing, with the exception of the SCS level network routing which is dependent upon the connection topology of SCS's.

2.2 Functional Decomposition

2.2.1 Introduction

As stated in the Introduction to Section 2.1, the purpose of this section is to discuss the 'column' functions for the Feasibility Development Model. Each column function will be decomposed in terms of the row functions considered in Section 2.1 necessary to the effective implementation of the column function. The column functions to be addressed in this section are:

1. VSQC - Voice Service Quality Control
2. DSQC - Digital Service Quality Control
3. BBSA - Baseband Signal Analysis
4. WBSA - Wideband Signal Analysis
5. SDCA - Switch Data Collection/Analysis
6. OCRI - Operator Control and Report Interface
7. FIAC -Fault Isolation and Analysis Coordination
8. SSCI - Station to Station Communicatins Interface
9. DBMS - Data Base Management Service

As previously mentioned, these functions are dependent upon the equipments and reporting facilities available at a station. The row components indicated as requisite for a column function must all be present in order to implement the function; consequently, any later modularizations made in terms of the function plane must take these columnar dependencies into account. It is worth mentioning that the FIAC holds a distinguished position in the scheme. That is, every station must contain some aspect of FIAC regardless of the rest of the station configuration; otherwise there would be no station.

2.2.2 VSQC - Voice Service Quality Control

This function appears only in an MAS configuration, and then only if channel analog signals are broken-out at the station. The purpose of the function is to provide quantitative measures of the performance of the channel with respect to the quality of the reproduced signals. The function may also detect failures within the system which do not appear at the BBSA level of analysis. In most cases, however, fault detection is a corollary to a BBSA detected fault. The primary use of the performance measures provided by this function is to predict failing components, where a failure may not be a loss of signal but a loss of the information carrying capacity of the channel.

The row components that make up the VSQC function are:

1. DC(VSQC) - a unique VSQC Data Collection component
2. DR(VSQC) - a unique VSQC Data Reduction component
3. DA(mf) - the common Data Assessment component (threshold/ alarm and trend)
4. DBR - the common Data Base Request component
5. R(VSQC) - a unique VSQC reporting component
6. SC - the common Scheduling component
7. CCI(mf)

Refer to Figure 2-4a for the control flow of this function.

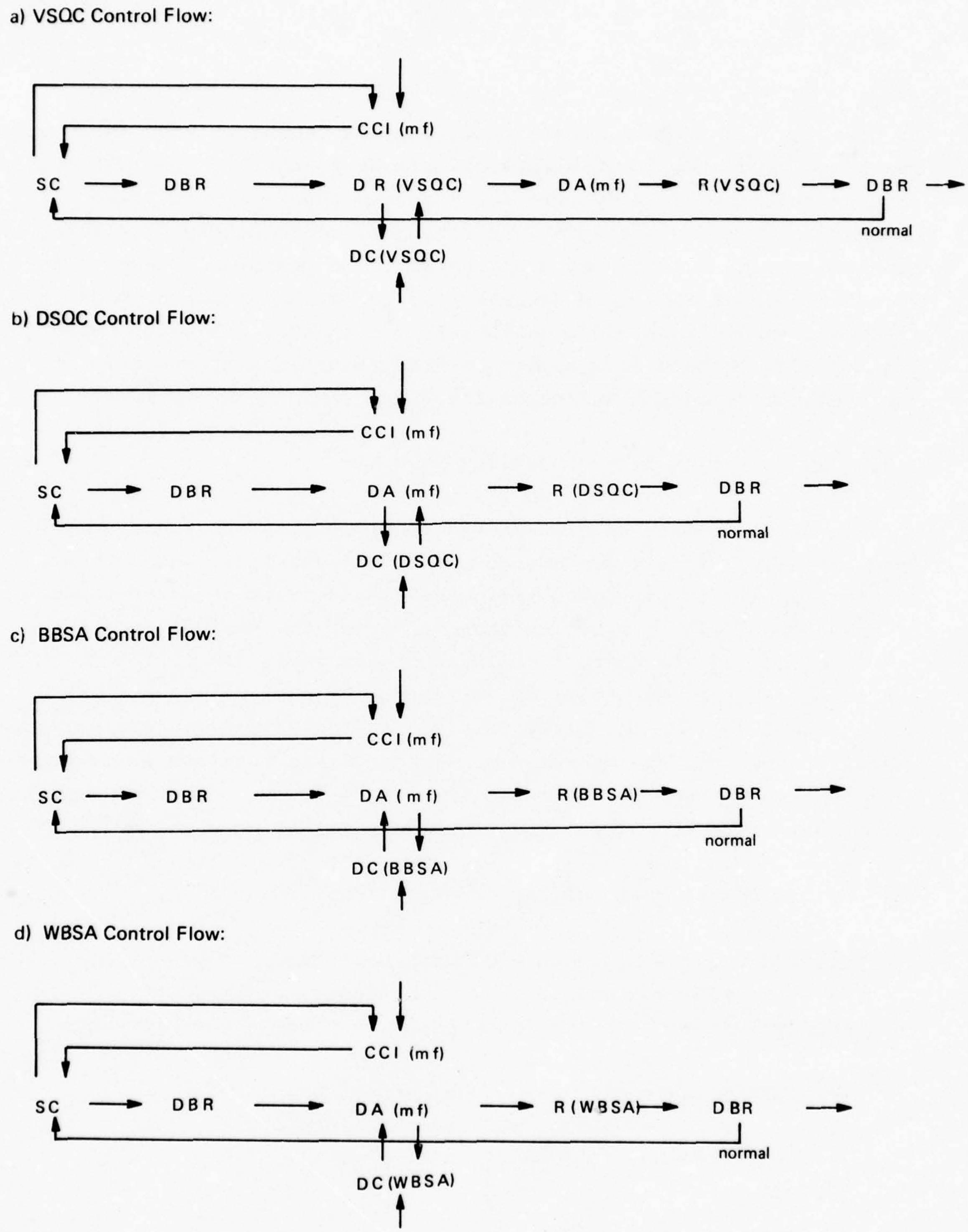


Figure 2-4
Control Flows of Equipment Measurement Functions

2.2.3 DSQC - DIGITAL SERVICE QUALITY CONTROL

This function appears only in an MAS configuration, and then only if channel level digital signals are borken-out at the station. The purpose of the function is to provide quantitative measures of the channel performance with respect to error characteristics of the information carried on the channel. This function may be used with either wideband digital or analog transmission facilities. As with VSQC, the primary indications provided by this function are degradations in the performance of equipments supporting the channels; however, this function may be of use in isolating faults at the group or supergroup level within the multiplex and transmission equipments.

The row components that make up this function are:

1. DC(DSQC) - a unique DSQC Data Collection component
2. DA - the common Data Assessment component (threshold/ alarm and trend)
3. DBR - the common Data Base Request component
4. R(DSQC) - a unique DSQC reporting component
5. SC - the common scheduling component
6. CCI(mf)

Refer to Figure 2-4b for the control flow of this function.

2.2.4 BBSA - BASEBAND SIGNAL ANALYSIS

This function appears at those MAS level stations which maintain analog multiplex and transmission facilities. The purpose of this function is to monitor performance of the equipments to the group level and to provide alarm detection within the transmission and multiplex equipments. Through the hard alarms triggered by this function,

immediate fault detection is provided; whereas the VSQC and DSQC functions may be thought of as pertaining more to the logical channel performance. Because the transmission equipments associated with analog multiplex facilities are different from those in the digital multiplex case, this function includes the transmission aspect of the function rather than separating these into two column functions.

The row components necessary for the implementation of BBSA are:

1. DC(BBSA) - a unique BBSA Data Collection component
2. DA - the common Data Assessment component (threshold/alarm and trend)
3. DBR - the common Data Base Request component
4. R(BBSA) - a unique BBSA Reporting component
5. SC - the common scheduling component
6. CCI(mf)

Refer to Figure 2-4c for the control flow of this function.

2.2.5 WBSA - Wideband Signal Analysis

This function is responsible for the monitoring of wideband digital multiplex and transmission facilities. It will be installed at those MAS level stations which maintain wideband digital multiplex and transmission equipments. This function serves the same purpose as the BBSA expect that the equipment complement is digital instead of analog.

The row components necessary for the implementation of the WBSA are:

1. DC(WBSA) - a unique WBSA Data Collection component
2. DA - the common Data Assessment component (threshold/alarm and trend)
3. DBR - the common Data Base Request component

4. R(WBSA) - a unique WBSA Reporting component
5. SC - the common scheduling component
6. CCI(mf)

Refer to Figure 2-4d for the control flow of this function.

2.2.6 SDCA - SWITCH DATA COLLECTION/ANALYSIS

This function is responsible for the collection and analysis of switch traffic data throughout the hierarchy (MAS, NCS, SCS). This function will be present at each MAS level station which has responsibility for a switch (AUTOVON or AUTODIN), at each NCS level station which has at least one MAS level station providing switch traffic data, and at each SCS level station which has at least one NCS level station providing switch traffic data and analysis. The purpose of this function is to provide statistics and raw data as necessary ultimately to ACOC and DCAOC levels of the SYSCON hierarchy. The component profile for this function is distributed throughout the three lowest levels of the hierarchy. The primary collection of information is accomplished at the MAS level; reduction and analysis components may be performed at the NCS and SCS levels prior to forwarding up the hierarchy.

The MAS level row components that implement this function are:

1. DC(SDCA) - a unique SDCA Data Collection component
2. DBR -- the common Data Base Request component
3. R(SDCA) - a unique SDCA Reporting component
4. CCI(SDCA) - the command and control component
5. DA(SDCA) - the data assessment component

Refer to Figure 2-5a for the control flow of this function.

Since this function is 'event' driven (by the arrival of a message), there is no need for the SC component.

SDCA Control Flow:

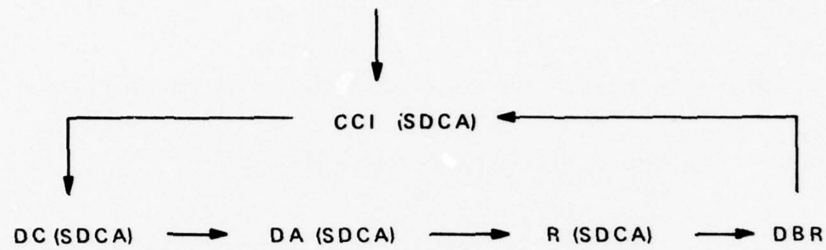
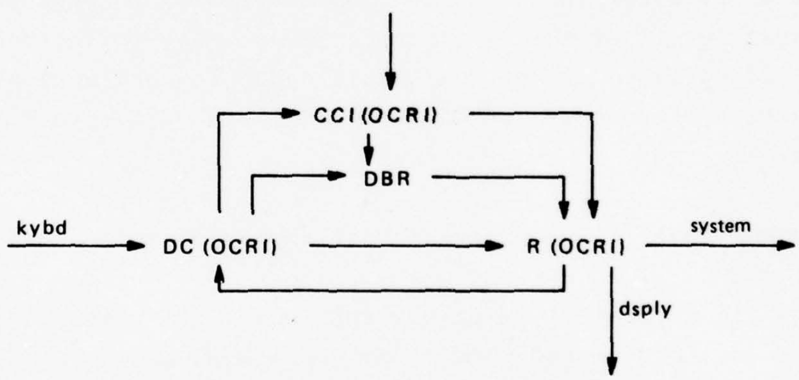


Figure 2-5

a) OCRI Control Flow:



b) FIAC Control Flow:

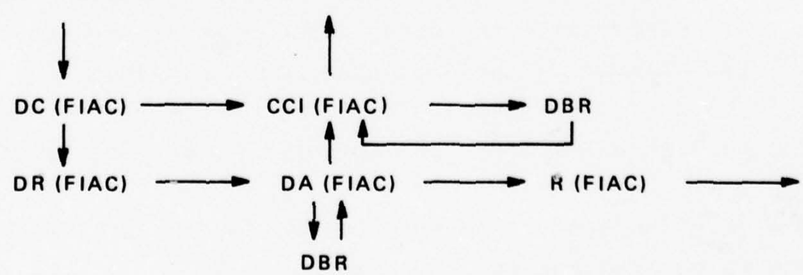


Figure 2-6

At the NCS and SCS levels the row components required are:

1. DC(SDCA) - a unique SDCA Data Collection component
2. DR(SDCA) - a unique Data Reduction component
3. DA - the common Data Assessment component (threshold/ alarm and trend)
4. DBR - the common Data Base Request component
5. R(SDCA) - the unique SDCA Reporting component
6. CCI(SDCA)

Refer to Figure 2-5b for the control flow of this function.

2.2.7 OCRI - Operator Control and Report Interface

The purpose of this function is to provide the support for operator interaction with the components at a station. The function is required at all manned stations. A description of the characteristics of this function follows.

The operator interface will require a keyboard entry and display output device. The specialized function key approach available on AN-GYM-13(v) is amenable to simulation on the display device. The hardcopy requirements may be satisfied by a 300 LPM or 600 CPS device. Hardcopy is required for logs of transactions involving the operator.

The operator interface serves three purposes:

1. Command interpretation to the Feasibility Development Model
2. Report generation function

3. Site-to-site operator communication facility

The requirements of command interpretation are that it enable both experienced and novice controllers to have complete access to the Feasibility Development Model. Access to the Model is defined as the ability to command specific sequences of measurement and assessment activity to be undertaken in the Model for whatever purposes the controller may have (typically fault isolation). To satisfy both ends of the operator experience spectrum, the specific interface must take into account that 1) a novice will not be expected to know the 'jargon' of tech control, 2) the interface should be of value as an instructional aid to the novice, and 3) the experienced controller will prefer to communicate with the Model in as concise a fashion as possible. An operator language suitable for the experimental FDM simulation system is described in Chapter 7.

The report generation function must support the operator entry of data into the predefined report formats discussed in the section on reporting. These formats should be displayed to the operator as a form to be filled in by the operator at the keyboard entry device. The interface must account for the requirements of data validation and necessary logging.

The site-to-site operator communication facility will serve as the order wire between sites. Its primary use will be as a coordination facility for O&M functions between sites, such as fault isolation. The facility should provide the controller with the ability to send intersite communications by addressing such communications either in terms of meaningful location names or in terms of familiar controller names.

The row function components involved with OCRI are:

1. DC(OCRI) - a unique Data Collection component for OCRI
2. R(OCRI) - a unique Reporting component for OCRI
3. CCI(OCRI) - a unique Command and Control Interpreter
4. DBR - The Data Base Request component

Refer to Figure 2-6a for the control flow of this function.

2.2.8 FIAC - Fault Isolation and Analysis Coordination

This function constitutes one of the primary capabilities of the lower three levels of the hierarchy. The responsibility of this function is to accept the event reports of functions VSQC, DSQC, BBSA, and WBSA, to analyze these reports for fault indications and to then initiate necessary isolation procedures to resolve to the equipment level the location of the fault. It is then the responsibility of the controllers to reconfigure and repair as necessary.

The need for this function is seen in the impracticality of placing a hard alarm indicator at every reasonable failure point, and the fact that a failure in one equipment can appear as a failure in some other distant equipment. The latter consideration suggests that in general there will be multiple fault 'images' generated within and among MAS sites; thus FIAC must be able to correlate these multiple fault images at the highest level (NCS or SCS) necessary to resolve the images to one casual failure. Conceptually, the correlation procedure is recursive throughout the three levels, as discussed in the component decomposition below.

The FIAC is itself distributed along the hierarchy. The row component decomposition at the MAS level is:

1. DC(FIAC) - the FIAC Data Collection receives the event reports generated by R(VSQC), R(DSQC), R(BBSA), and R(WBSA).

2. DA(FIAC) - the Data Assessment component has several tasks to perform: a) it performs the correlation procedure masking out all but the 'earliest' fault image at the MAS level; b) isolating to the largest fault level (i.e., circuit, trunk, or baseband); c) determines measurements necessary to confirm and isolate faults within the MAS.

3. DR(FIAC) - determines for each event report whether it represents a fault.

4. CCI(FIAC) - interprets commands from NCS level FIAC for fault confirmation and issues commands to CCI(VSQC), CCI(DSQC), CCI(BBSA), and CCI(WBSA) as necessary.

5. R(FIAC) - the Report component dispatches an event report to the responsible NCS DC(FIAC) as determined by the 'most' significant fault from the DA(FIAC). This report is necessary even if the fault can be directly isolated to this MAS so that the NCS and SCS may perform fault image correlation.

6. DBR - this component is used to access the connectivity information necessary to determine which event reports are 'earlier' than others.

Refer to Figure 2-6b for the control flow of this function.

The NCS and SCS levels are decomposed along the same lines as the MAS level FIAC; thus realizing the recursive nature of the isolation and analysis procedure.

The strategies to be employed in DA(FIAC) may be found in the CSC report Fault Isolation Algorithm, FS-4952-0200B.

2.2.9 SSCI - Station to Station Communication Interface

This function serves to route and switch both command/control messages and report messages between stations. It is required at all levels (MAS, NCS, and SCS). Report messages arrive at a station from stations below it in the hierarchy. Command and control messages leave a station for ones below it in the hierarchy and consequently these messages arrive at a station from the one above it. In the case of SCS level stations, both types of traffic may be leaving and

arriving from stations at the same level. For example, a report message from a MAS R(SDCA) would be routed by the MAS SSCI to the responsible NCS SSCI which would in turn route the report to the NCS DC(SDCA) for action. Hence, it is the responsibility of the SSCI to perform the functional address to physical address mapping.

If the interstation communications link is found to be not functioning at some point, R(SSCI) forwards a message to OCRI if it exists, and buffers outgoing message traffic by the use of DBR. The buffering is done in two classes: a) SDCA reports and b) all other traffic. The SDCA report buffer is a revolving area sized as discussed in Section 2.1.4 to hold the latest two hours of traffic reports.

The row components employed by SSCI are:

1. CI - the Communications Interface
2. DBR - the Data Base Request component
3. R(SSCI) - the Report component unique to SSCI necessary only if OCRI is present

Refer to Figure 2-7a for the control flow of this function.

2.2.10 DBMS - Data Base Management Service

This function is responsible for effecting the commands forwarded to it by the DBR components of the other functions. DBMS is of course responsible for maintaining the integrity of the data base under its control. Integrity in this context is ensuring that update and add requests are properly authorized. For example, SDCA should not be updating information under the control of BBSA, and OCRI at an NCS should not be updating the connectivity data base under the control of FIAC. The latter update should originate from the controlling SCS level and carry with it the appropriate authorization to perform the update.

The CCI(DBMS) is the component responsible for the integrity checks of the sort discussed above. Valid requests are then scheduled, with FIAC requests receiving high priority. The DBC does all the work of performing the mapping of requests to physical location and locking of records. A record is locked for example by a measurement function when it retrieves a record prior to monitoring, since the updated record must be returned and placed back in the data base. R(DBMS) forwards the results of requests to the originator.

The row components of the DBMS are:

1. CCI(DBMS) - Command and Control Interpreter
2. SC - Scheduling component
3. DBC - the Data Base Component
4. R(DBMS) - the DBMS Reporting component

Refer to Figure 2-7b for the control flow of this function.

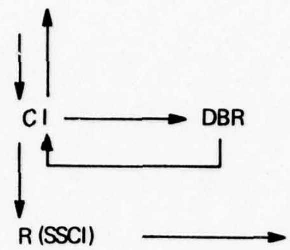
2.2.11 Summary

Section 2.1 has discussed the requirements of a number of functions in terms of the row components of these functions. The information presented in Section 2.1 is intended to provide input to the Feasibility Development Model Design Task 11, (Ch.6) in terms of the microprocessor requirements with regard to size and speed.

Section 2.2 has discussed the decomposition of the functions and indicated the functions which are required and the components necessary to implement them. This information will be integrated in Chapter 3 concerning module definition.

Figure 2-8 summarizes the dependencies of functions with respect to each other, and Figure 2-9 summarizes the row function/column function plan as developed in this report.

a) SSCI Control Flow:



b) DBMS Control Flow:

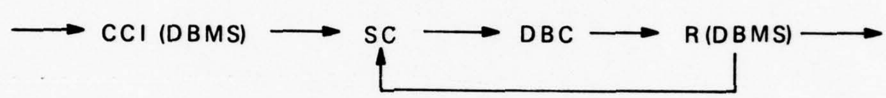


Figure 2-7

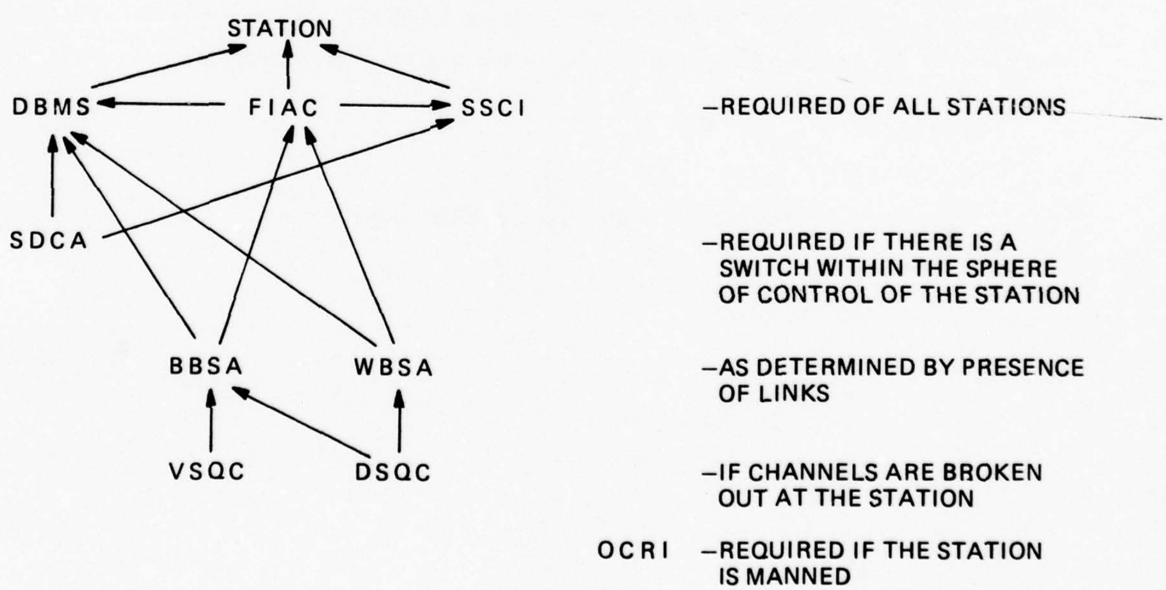


Figure 2-8
Function Dependencies

	VSQC	DSQC	BBSA	WBSA	SDCA	FIAC	DBMS	OCRI	SSCI
DC	1	2	3	4	5	6		7	
DR	1					2			
DA	1	1	1	1	2	3			
DBR	1	1	1	1	1	1		1	1
R	1	2	3	4	5	6	7	8	9
CCI	1	1	1	1	2	3	4	5	
SC	1	1	1	1			1		
CI									1
DBC							1		

NOTE: The numbers in each row indicate the unique versions of the row component which were identified in the report. Thus, there are 7 different DC components and 3 DA components for example.

Figure 2-9.
Row Function/Column Function Plane

3. MODULE DEFINITION

3.1 Introduction

A module, in the context of this report, is an implementation entity. As such, a module will have hardware, software, and possibly firmware components. In order to achieve a useful modular implementation, there must be a partitioning of the problem which identifies both independent and redundant aspects of the problem functionally. Such a partitioning with respect to the MSCDM problem was developed in terms of a function plane in Chapter 2 on Functional Analysis and Decomposition. The task at hand is to consider how the components identified in the previous chapter may be organized with respect to modules as implementation entities.

There are several objectives to be gained by employing the approach of modularization to the implementation of a system. From the hardware point-of-view, it is desirable to have a small number of distinct replaceable components. This reduces overhead in spare parts inventory by reducing the number of types of parts, since one part may be used for a number of different functions, and a reduction in both design cost and manufacturing cost is realized since a large number of standard parts may be produced rather than a fewer number each of many parts. Further, the cost of training personnel and the MTTR are both reduced since the field personnel in effect have to know less hardware. Within the context of this study, hardware modules will be constructed around a set of one or two off-the-shelf microprocessor components. This fact implies an evaluation criterion with respect to the architecture that the cost of interconnecting such hardware modules must be in proportion to the cost of the modules. That is, the throughput capacity of the interconnection architecture should be of the same order of magnitude as the throughput capacity of the processors being interconnected. This may preclude the use of such approaches as switch interlocks which have been used in the past to construct multiprocessor systems, since the cost of such an interconnection approach would be over-powered with respect to the use

of inexpensive microprocessors and with respect to the problem requirements. Another objective in terms of the hardware modularization is that the modules should represent reasonable size or capacity with respect to the incremental changes in monitored equipments installed at stations so that the monitoring system may be effectively modified with corresponding increments of hardware. Further, these changes in configuration should be amenable to field installation rather than requiring depot level installation.

With respect to software and firmware the major benefits are accrued in the design and implementation phases of a system. By employing the techniques of information hiding [Parnas,72], software components may be constructed which have several useful attributes: they may be developed, debugged and modified independently, and they may be interconnected in a plug compatible manner by the use of well defined interfaces. The information which is to be hidden are the design decisions which relate to the internal functioning of a module such as data structures and character codes. In this approach modules may not necessarily correspond to big steps in the processing as in a purely sequential decomposition but to functional capabilities used at several points of the total processing. New functional requirements may be introduced into the family of software modules by considering only the pre-existing system software interface conventions. As with the hardware modules the software modules should be amenable to field installation. This implies that there must be a system utility which performs the plugging together of modules to construct a functional capability (e.g. loader programs).

The function plane partitioning of the previous report immediately suggests two dimensions along which the definition of modules may be developed:

1) the row functions represent modularity in the construction of the column function modules; and

2) the column functions represent a modularity in terms of station capabilities.

Thus, the entire development will be presented along the four dimensions so far identified: hardware, software, row modules, and column modules.

The row functions can be viewed as representing a fine grain view of the functional capabilities which are used to implement stations. It is through the row function partitioning of the problem that the redundant functional characteristics of the problem are identified. This redundancy is seen as the appearance of a specific component two or more times in a particular row of the function plane.

DC - Data Collection Components:

These components are the major source of unique hardware requirements. Since they represent the interfaces of the monitoring system to the monitored equipments. From a hardware point-of-view these modules may be considered to be I/O devices. From the software point-of-view these components play the familiar role of device handlers in a classical operating system.

DR - Data Reduction Components:

The function plane identifies two such components. The purpose of the data reduction component is to pre-process collected data to provide parameters which may be meaningfully processed by the data assessment module applicable to the function.

DA - Data Assesment Components:

These components are responsible for the analysis of measured parameters to determine if an event, usually a fault, has occurred which should be reported.

DBR - Data Base Request Component:

This is a single module which performs the task of communicating data base requests to the DBMS. The requests are of the form add, delete, and update. All column modules require the DBR since all permanent storage in the system is managed through the DBMS. This function serves primarily to hide the information concerning the structure and maintenance of the data base from those modules which must interact with the database.

R - Reporting components:

There is a separate reporting module for each column module that has been defined within the system. The reporting modules format the output of their respective column modules either for OCRI display or to be forwarded to higher level modules, such as FIAC. In the case of OCRI reporting, these modules hide the information concerning the formating of the controller display device. For FIAC reporting, these modules create standard event reports which may be interpreted by the appropriate FIAC. In the case of the R(OCRI), this module will include the hardware necessary to connect the controller display device to the system. It is also responsible for calling on the DBMS via the DBR module to provide the display forms necessary for standard controller reports.

CCI - Command and Control Interpreter Components:

These components are responsible for the interpretation of system commands directed toward the various column modules. These modules are software only. They hide the system inter-module command protocol from the other row modules which are used to construct the various column modules.

SC - Scheduling Component:

There is one common scheduling module which is a pure software module. The function of this module is to maintain the queue of interpreted requests for the column function module of which it is a part. It responds to the several commands discussed in the previous report. These commands and the associated requests are passed to SC from the CCI associated with the particular column function module.

CI - Communications Interface Component:

This is a unique module which includes the hardware necessary to interface stations together, as well as the inter-station routing software.

DBC - Data Base Component:

This is a unique module which is the critical component of the DBMS column module. It includes the hardware necessary to store the data base information as well as the software which will manage the device and mapping of logical record identifiers to physical location.

The column function modules represent the implementation of station capabilities which are necessary to construct stations. The logical level interface between column function modules, is accomplished by a message transfer mechanism. The internal format of the information portion of the message will be module dependent. The routing control or addressing portion of the message will be standard through-out the system. Since the information field of the message is interpreted by the destination module (by the CCI), it is easy to extend the system interface protocol to encompass new modules, with new information formats. Messages provide a useful degree of information hiding, in that, the system interconnection facilities are concerned only with the transmission and delivery of messages, and not with the particular information formats of the different column function modules.

Messages also provide a uniform method of intra as well as inter station coupling of modules. A parameter calling sequence type of interface between column function modules is inappropriate since, it would introduce complexity due to the differing number and types of parameters for the various modules and due to the asynchronous execution requirements of the modules.

It is through the column function partitioning that the independent functional characteristics of the problem have been identified and employed in the system design. The independence is seen for example in the separation of the WideBand Digital monitoring from the Base-Band(analog) monitoring. In other words, a station with only wide band digital facilities need install only the WBSA module, no part of the BBSA is necessary to the WBSA.

The coupling relations, with respect to the modularization discussed earlier in this section, concern how 'close' the various modules must be to one another in a given station configuration in order to achieve the performance requirements which were discussed in Chapter 2 on Functional Analysis and Decomposition.

There are two primary relations to be considered:

1) the relationships of the row modules used to construct a column module to one another.

2) the relationships amongst the column modules which are to be used to construct stations.

In general, the row modules used to construct a column module are executed sequentially; that is, only one module can be active at a time. For example, DC(VSQC) must wait until DBR has retrieved the data base entry for the channel being monitored, before it can begin the collection of data. The performance requirements of VSQC, for

example, imply that the data points collected by DC(VSQC) must be rapidly available to DR(VSQC). These considerations suggest that the row modules should be tightly coupled when implementing a column module. Further, the tight coupling of row modules lends itself nicely to packaging the row modules as a single component for construction purposes. A straight forward way of achieving the tight coupling is to assign a column module, thus constructed, to a processor. Notice that the intra-column module flow of control is amenable to a subroutine calling protocol due the sequential character of execution of the row modules. It will be seen in Chapter 6 of this report that in general each of the identified column modules may be mapped directly to single micro-processors. In some cases more than one column module may be mapped to a single processor if either the sum of the processing loads of the modules are smaller than the thoughput capacity of the processor or if the modules are in most cases synchronous in execution (i.e. either one or the other are in execution but usually not both).

The column module interactions, however, are largely asynchronous in that each module is independent in function of the others. For example WBSA may function in parallel with SDCA, and so forth. Secondly, due to the independence of function, there is little requirement for data sharing, and in the case of DBMS, FIAC, and SSCI there is a convergence of interconnection requirements in that most column modules have occasion to participate in transfers of information between the above mentioned column modules. In effect, DBMS, FIAC, and SSCI are the central nodes in star networks of interconnections with the other column modules. The data transfers between column modules may be characterized as bursty, in that they tend to be relatively infrequent transfers of several hundred bytes per transfer. These considerations suggest that the inter-column module coupling requirement is loose with respect to intra-column module requirements.

The remaining sections of this chapter consist of the specifications for the column modules in terms of their respective row components and input and output requirements.

3.2 VSQC - Voice Service Quality Control:

This module is to be implemented at sites which have 4 kHz analog voice channels broken-out at the site. The module performs performance assessment of voice channels for the purpose of detecting degrading performance and assisting in fault isolation tasks on site equipments.

Inputs:

Channel signal level is input to DC(VSQC) and is used as the basis of performance assessment. Channel signal level is input as 12 bit A/D value at a sampling rate of 12.8 kHz. Commands to VSQC from the system are accepted by the CCI(mf) which is the system input interface to the module. Database records requested by the DBR are also accepted by the CCI(mf) which will be in control of the module while a DBR initiated request is outstanding.

Outputs:

The channel select and bridging commands to the channel interface are issued from DC(VSQC). System outputs consist of DBR requests directed to DBMS, and reporting output directed to both OCRI and FIAC.

Row Components:

1) DC(VSQC):

This component comprises the channel interface hardware which accepts channel select and bridging commands and returns A/D channel signal

level values upon request. This component will collect 128 channel signal level values in a 10 milisecond interval. The channel selection and bridging information are provided as parameters from the DR (VSQC) which initiates DC (VSQC).

2) DR (VSQC): This component is initiated after the data base record for the channel to be assessed has been retrieved by DBR from the DBMS. DR (VSQC) will maintain control of VSQC until all data has been collected and reduced. The collection of data consists of calling DC(VSQC) at 100 millisecond intervals until 32 sets of 128 sample points have been collected. Since the collection of sample points requires only 10 milliseconds, DR(VSQC) performs reduction operations during the intervening 90 milliseconds. The operations consist of computing the peak and average power and FFT of the previous 128 sample points. At the conclusion of accumulating 32 sets of data, the average spectrum and the channel peak and average power values are computed for the total sample period. The results are placed in the transfer buffer and control is passed to DA(mf).

3) DA(mf):

The Data Assessment component examines the transfer buffer to determine the module type (which is VSQC) in order to set its operating mode. In the case of VSQC, DA(mf) retrieves the data to be trended directly from the transfer buffer rather than calling a DC component as this component does in other measurement function modules. The trending and thresholding operations are performed using the previous state information held in the transfer buffer. The updated state information is placed in the transfer buffer and control is passed to R(VSQC).

4) R(VSQC):

This component examines the transfer buffer for event conditions which indicate that a report needs to be generated for FIAC. If so

indicated a report is issued to FIAC. Next, any OCRI display requests are formatted and issued to OCRI for display. Control is then passed to DBR to update the database.

5) DBR:

This component is called by SC to initiate the monitoring of a channel. The request in this case is for the retrieval of the database record of the channel to be monitored. The next component control indicates DR(VSQC) so that the completion of the DBR activity will result in control being passed to DR(VSQC) to begin the collection and reduction process. DBR is called by R(VSQC) after all reports have been issued so that the updated transfer buffer may be sent with an update request to the DBMS. Next control then passes to SC.

6) SC:

The scheduling component is invoked either by the CCI(mf) if the module was in the IDLE state or by DBR at the completion of monitoring of a channel. If control came from DBR the transfer buffer is checked for the WAIT-FOR-FIAC indicator. If this indicator is present, control is relinquished to CCI(mf); otherwise, the schedule command request queue is examined for requests left by CCI(mf), lastly the monitor scanning request queue is examined for the next scanning operation to be performed. If both queues are null, control is passed to CCI(mf) and the state becomes IDLE.

7) CCI(mf):

This component may be entered either explicitly from SC, or implicitly from DBR when a request is initiated. The receive buffers are examined for system commands which have arrived. Commands are interpreted and entries made in the schedule command request queue or the transfer buffer for OCRI display requests. The arrival of the results of DBR requests cause the DBR to be reentered.

3.3 DSQC - Digital Service Quality Control:

This module will be implemented at those sites which contain digital data channels broken-out at the site. The DSQC module is capable of assessing the performance of digital data channels for the purpose of detecting degrading performance of these channels with respect to increasing error rates, and to assist in fault isolation tasks on site equipments.

Inputs:

Pseudo error, bit error, and block error rates are input to DC(DSQC) and form the basis of performance assessment on digital channels. System commands directed to DSQC are accepted by the CCI(mf) component which is installed in the DSQC module. The results of DBR requests are also accepted by the CCI(mf) which is in control while DBR requests are outstanding.

Outputs:

The channel select commands to the digital channel monitoring interface are issued from DC(DSQC). DC(DSQC) also issues commands which cause the input of the pseudo, bit, and block error rate parameter values for the channel selected. Outputs directed to the system include DBR requests directed to the DBMS module, and Reporting outputs directed to both OCRI and FIAC system modules.

Row Components:

1) DC(DSQC):

This component includes the digital channel monitoring interface hardware which accepts channel select and input parameter commands and returns the current parameter values requested. This component

is called by DA(mf) and is passed the channel designation of the channel to be monitored. The parameter values are input and placed in the transfer buffer, control is then passed back to DA(mf).

2) DA(mf):

The DA(mf) component receives control after the database record for the channel to be monitored has been retrieved by the DBR. The transfer buffer is examined and will indicate that it is the responsibility of DA(mf) to call DC(DSQC) to place the channel parameter values in the transfer buffer. The trending and thresholding operations are performed upon the current parameter values using the previous state information held in the transfer buffer. The updated state information is placed in the transfer buffer and control is then passed to R(DSQC).

3) R(DSQC):

This component examines the transfer buffer for event conditions which indicate that a report should be issued to FIAC. Next, any OCRI display requests are formatted and issued to OCRI for display. Control is then passed to DBR to update the database.

4) DBR:

This component receives control from SC to initiate the monitoring of a channel. The DBR formats a retrieval request to DBMS for the database record of the channel to be monitored. The next component control indicates DA(mf) so that at completion of the retrieval request control is passed to DA(mf) to collect and assess the channel parameters. DBR receives control from R(DSQC) after all reports have been issued so that the updated transfer buffer may be sent with an update request to the DBMS. Next component control then passes to SC.

5) SC:

The Scheduling component receives control from either the CCI(mf) if the module was in the IDLE state or from DBR at the completion of channel monitoring. If control came from DBR the transfer buffer is examined for the WAIT-FOR-FIAC indicator. If this indicator is present control is returned to CCI(mf) to await the FIAC response; otherwise, the scheduler command queue is examined for requests left by CCI(mf) which indicate changes in the scanning flow. Lastly, the monitor scanning request queue is examined for the next scanning operation to be performed. If both queues become empty, control is returned to CCI(mf) and the module state becomes IDLE.

6) CCI(mf):

This component receives control either explicitly from SC when the module becomes IDLE or implicitly from DBR when a DBMS request is initiated. The receive buffers are examined for system commands which have arrived. Commands are interpreted and entries made in the scheduler command request queue or the transfer buffer for OCRI display requests. The arrival of the results of DBR initiated requests cause the DBR to be reentered. FIAC responses appear in the receive buffers as system commands and are handled accordingly.

3.4 BBSA - Base Band Signal Analysis:

This module is implemented at those sites which are responsible for at least one analog link equipment complement consisting of receiver, transmitter and optional multiplexer/demultiplexer. The BBSA module is able to assess the performance of the link equipments for the purpose of detecting failed elements and assisting in fault isolation tasks on site equipments.

Inputs:

DC(BBSA) has the ability to collect digitized values for:

transmitter percent modulation
transmitter frequency deviation
relative transmitter power
receiver AGC voltages
receiver IF output
multiplex baseband levels
multiplex pilot levels

These parameters form the basis of link assessment. In addition transmitter, receiver, and multiplexer alarms are input as vectored interrupts to the BBSA processing element. System commands directed to BBSA are accepted by the CCI(mf) component which is installed in the BBSA module. The results of DBR requests are also accepted by the CCI(mf) which is in control while DBR requests are outstanding.

Outputs:

Link equipment select and parameter request commands are issued from the DC(BBSA) to the analog link monitoring interface hardware. Outputs directed to the system originate in the DBR component where they are directed to the DBMS to perform retrieval and update operations, and from R(BBSA) where the outputs may be directed to both FIAC in the case of event reports and to OCRI in which case the output is a formatted display.

Row Components:

1) DC(BBSA):

This component of BBSA includes the analog link monitoring interface hardware which accepts the link equipment select and parameter request commands and returns the digitized parameter values requested.

This component receives control from the DA(mf) and is passed the parameter request list. The digitized parameters are requested and placed in the transfer buffer, control is then returned to DA(mf).

2) DA(mf):

The DA (mf) component receives control after the database record for the link to be monitored has been retrieved by the DBR. The transfer buffer is examined and will indicate that DC(BBSA) must be called to place the parameter values for the link in the transfer buffer. The trending and thresholding operations are performed upon the parameter values using the previous state information held in the transfer buffer. The updated state information is placed in the transfer buffer and control is then passed to R(BBSA).

3) R(BBSA):

This component examines the transfer buffer for event conditions which indicate that a report should be issued to FIAC. Next, any OCRI display requests are formatted and issued to OCRI for display. Control is then passed to DBR to update the database.

4) DBR:

This component receives control from SC to initiate the monitoring of a channel. The DBR formats a retrieval request to DBMS for the database record of the channel to be monitored. The next component control indicates DA(mf) so that at completion of the retrieval request control is passed to DA(mf) to collect and assess the channel parameters. DBR receives control from R(DSQC) after all reports have been issued so that the updated transfer buffer may be sent with an update request to the DBMS. Next component control then passes to SC.

5) SC:

The Scheduling component receives control from either the CCI(mf) if the module was in the IDLE state or from DBR at the completion of channel monitoring. If control came from DBR the transfer buffer is examined for the WAIT-FOR-FIAC indicator. If this indicator is present control is returned to CCI(mf) to await the FIAC response; otherwise, the scheduler command queue is examined; or requests left by CCI(mf) which indicate changes in the scanning flow. Lastly, the monitor scanning request queue is examined for the next scanning operation to be performed. If both queues become empty, control is returned to CCI(mf) and the module state becomes IDLE.

3.5 WBSA - Wide Band Signal Analysis:

This module is implemented at those sites which contain wide band digital link equipments. The WBSA module is capable of assessing the performance of wide band digital link equipment with the purpose of detecting degrading conditions related to the link equipments and assisting on the fault isolation tasks on site equipments.

Inputs:

DC (WBSA) has the ability to collect digitized values for:

- Transmitter Percent Modulation
- Transmitter frequence deviation
- Relative transmitter power
- Receiver AGC voltages
- Receiver IF output
- Multiplex Pseudo Error Rate

These parameters form the basis of link assessment. In addition transmitter, receiver, and multiplexer alarms are input as vectored interrupts to the WBSA processing element. System commands directed to WBSA are accepted by the CCI(mf) component which is installed in the WBSA module. The results of DBR requests are also accepted by the CCI(mf) which is in control while DBR requests are outstanding.

Outputs:

Link equipment select and parameter request commands are issued from the DC(WBSA) to the digital link monitoring interface hardware. Outputs directed to the system originate in the DBR component where they are directed to the DBMS to perform retrieval and update operations, and from R(WBSA) where the outputs may be directed to both FIAC in the case of event reports and to OCRI in which case the output is a formatted display.

Row Components:

1) DC(BBSA):

This component of WBSA includes the digital link monitoring interface hardware which accepts the link equipment select and parameter request commands and returns the digitized parameter values requested. This component receives control from the DA(mf) and is passed the parameter request list. The digitized parameters are requested and placed in the transfer buffer, control is then returned to DA(mf).

2) DA(mf):

The DA(mf) component receives control after the database record for the link to be monitored has been retrieved by the DBR. The transfer buffer is examined and will indicate that DC(WBSA) must be called to place the parameter values for the link in the transfer buffer. The trending and thresholding operations are performed upon the parameter values using the previous state informatin held in the transfer buffer. The updated state information is then placed in the transfer buffer and control is then passed to R(BBSA).

3) R(RBSA):

This component examines the transfer buffer for event conditions which indicate that a report should be issued to FIAC. Next, any OCRI display requests are formatted and issued to OCRI for display. Control is then passed to DBR to update the database.

4) DBR:

This component receives control from SC to initiate the monitoring of a channel. The DBR formats a retrieval request to DBMS for the database record of the channel to be monitored. The next component control indicates DA(mf) so that at completion of the retrieval request control is passed to DA(mf) to collect and assess the channel parameters. DBR receives control from R(DSQC) after all reports have been issued so that the updated transfer buffer may be sent with an update request to the DBMS. Next component control then passes to SC.

5) SC:

The scheduling component receives control from either the CCI(mf) if the module was in the idle state or from DBR at the completion of channel monitoring. If control came from DBR, the transfer buffer is examined for the wait-for FIAC indicator. If this indicator is present, control is returned to CCI(mf) to await the FIAC response; otherwise, the scheduler command queue is examined for requests left by CCI(mf) which indicate changes in the scanning flow. Lastly the monitor scanning request queue is examined for the next scanning operation to be performed. If both queues become empty, control is returned to CCI(mf) and the module state becomes idle.

6) CCI(mf):

This component receives control either explicitly from SC when the module becomes idle or implicitly from DBR when a DBMS request is initiated. The receive buffers are examined for system commands which have arrived. Commands are interpreted and entries made in the scheduler command request queue or the transfer buffer for OCRI display requests. The arrival of the results of DBR initiated requets cause the DBR to be reentered. FIAC responses appear in the receive buffers as system commands and are handled accordingly.

3.6 SDCA - Switch Data Collection and Analysis:

This module wil be installed at those stations which contain at least one switch (e.g., AUTODIN or AUTOVON). This module receives switch traffic data generated by the switch and performs loading assessments on this data to detect switch equipment saturation conditions. The switch traffic data is forwarded the ACOC for analysis.

Inputs:

DC(SDCA) accepts switch traffic data on an event basis over 2400 baud connections. The data packets are 1024 bits in size with a nominal arrival rate of one packet every five seconds per switch. The switch traffic data packets form the basis of switch equipment saturation detection*. System commands directed to SDCA are accepted by the CCI(SDCA) component which is installed in the SDCA module. The results of DBR requests are also accepted by the CCI(SDCA) which is in control while DBR requests are outstanding.

---

*Switch traffic data packets are collected for the parameter groups listed in Sec. 2.2.2.5.

Outputs:

Outputs directed to the system originate in the DBR component where they are directed to the DBMS to perform retrieval and update operations, and from R(SDCA) where the outputs may be directed to OCRI in which case the output is a formatted display of the switch traffic data packet. R(SDCA) also directs a copy of the switch traffic data packet to SSCI for forwarding to ACOC.

Row Components:

1) DC(SDCA):

The Data Collection Component of SDCA includes a 2400 baud digital communications interface to each message/packet switch from which it must collect data. This component receives control from CCI(SDCA) and waits until a switch traffic data packet is received. The packet is placed in the transfer buffer for the switch from which the packet was received and control is passed to DA(SDCA) with the transfer buffer pointer as a parameter.

2) DA(SDCA):

The SDCA Data Analysis Component performs trending and threshold operations on the data in the transfer buffer indicated by the pointer passed to DA(SDCA) by DC(SDCA) and using the previous state information held in that transfer buffer. The updated state information including switch equipment saturation conditions is placed in the transfer buffer and control is passed to R(SDCA).

3) R(SDCA):

This component examines the transfer buffer for any event conditions which indicate that OCRI should be notified of switch saturation. R(SDCA) then issues any OCRI requested displays and forwards

the switch traffic data packet held in the transfer buffer to SSCI for transfer to ACOC. Control is then passed to DBR to update the database record for the switch.

4) DBR:

DBR receives control from R(SDCA) after all reports have been issued so that the updated trasnfer buffer may be sent with an update request to DBMS. Control is then passed to CCI (SDCA). Since the switch traffic data transfer buffer is always resident, DBR is not called to perform a retrieval operation.

5) CCI(SDCA):

This component receives control from DBR implicitly during the update and explicitly at the completion of the update. The receive buffers are examined for system commands which have arrived. The commands are interpreted and OCRI display requests cause entries to be made in the appropriate transfer buffers. The system idel request results in CCI(SDCA) retaining control until a system continue command is received. If SDCA is in the active state CCI(SDCA) passes control to DC(SDCA) to await the next switch traffic data packet.

3.7 DBMS - Data Base Management Service:

This module performs he data base maintenance functions required of SYSCON sites and is a required module in all site implementations. It is capable of initializing a site data base in accordance with the site configuration and performing retrieval, update, delete and add requests on that data base.

Inputs:

System ommands directed to the DBMS are accepted by the CCI(DBMS) component. Commands are directed to DBMS by the DBR components of other modules in the site. In addition requests for status displays may be issued by the OCRI module. Solicited inputs from the database device are accepted by the DBS.

Outputs:

System directed outputs originate with R(DBMS) and are the results of DBR requests from other modules. R(DBMS) also prepares and forwards OCRI requested status displays to the site OCRI. The DCS issues read, write and control commands to the database device.

Row Components:

1) DBC:

The Data Base Component includes the database device hardware interface and the storage medium for the database. This component receives control from the scheduler component and is passed a transfer buffer pointer containing logical record information and then translates the logical record identification into the physical location information necessary to control the database device. The operation is then initiated to the database device and the transfer buffer status is marked pending. Control is then passed to CCI(DBMS). When the database device request is complete control is returned to DBC and the transfer buffer status is marked as complete. Control is then passed to R(DBMS) to forward the results of the request.

2) R(DBMS):

The transfer buffer is first examined for event indications which require notifying the OCRI of a database device failure. Next the results portion of the transfer buffer are dispatched to the module address of the originating DBR. This address is found in the requestor field of the transfer buffer. If there are any outstanding OCRI status display requests noted these are formatted and issued to the OCRI. Control is then passed to SC.

3) SC:

The Scheduling Component receives control from either CCI(DBMS) if the module was previously idle, or from R(DBMS) at the completion of dispatching results. The database request queue is examined for the next request to be handed to DBC for processing. If the queue is found empty control is passed to CCI(DBMS) and the module status becomes idle.

4) CCI(DBMS):

This component receives control from SC when the module becomes idle or from DBC when a database device operation is initiated, in which case the module state becomes pending. Incoming requests are formatted into transfer buffers and DBC requests entered in the database request queue. If the state is pending and database device operation complete is noted, the DBC is reentered to complete the database request.

3.8 OCRI - Operator Control and Report Interface

This module performs the functions of interfacing to the site operator. The language interface will allow the operator to control the site, request status information concerning site and equipment performance, and to prepare site reports which must be forwarded to other sites such as the ACOC. Operator to operator messages are also supported by the OCRI.

Inputs:

Inputs directed to OCRI from other modules within the site system are accepted by CCI(OCRI). These inputs represent either solicited or unsolicited displays or the results of a DBR request. Inputs from the operator input device are accepted by DC(OCRI) on an event basis.

Outputs:

Status and control requests to other modules in the site system are issued by R(OCRI). These requests include the dispatching of site reports which are destined to leave the site via SSCI. DBR requests are output to retrieve operator report forms and to perform operator requested database operations. Output to the operator display device are performed by R(OCRI).

Row Components:

1) DC(OCRI):

This component includes the operator input device hardware. Operator input is either a command or a report input. In the case of a command input the DC(OCRI), which implements the operator language described in Chapter 7, syntaxes and translates the command into a system request. This request is placed in a transfer buffer to be dispatched by R(OCRI). If the module is in report state, the input is placed in a transfer buffer to be incorporated in the report by R(OCRI). If a command to enter report state is encountered or a database operation is requested control passes to DBR; otherwise, control is passed to R(OCRI).

2) DBR:

The DBR component is entered from DC(OCRI) to initiate the request found in the transfer buffer pointed to by the parameter to DBR or it is entered from CCI(OCRI) at the completion of a DBR request. Control then passes to R(OCRI).

3) R(OCRI):

This component is responsible for outputting commands to the system, controlling the operator display device, and preparing site reports based on operator input and forms contained in the site database. Control passes to R(OCRI) from one of three components: DC(OCRI), DBR, or CCI(OCRI). In all cases a transfer buffer pointer is passed which points to a transfer buffer which will control the execution of R(OCRI). At the completion of an operation R(OCRI) will return control to DC(OCRI) to wait for the next operator input.

4) CCI (OCRI):

This component is entered as an interrupt routine when an input from the system arrives, or while a DBR request is pending. The input will be prepared in a transfer buffer and control is passed to R(OCRI) or DBR if the input is the result of a DBR request.

3.9 FIAC - Fault Isolation and Control Coordination:

This component interprets event reports from measurement function modules and issues commands to measurement function modules or to other site FIAC modules for the purpose of isolating the equipment causing the detection of a fault condition. The FIAC also receives fault isolation requests from other site FIAC modules which are interpreted as within site measurement function module commands.

Inputs:

All inputs to FIAC originate within the system and are accepted by DC(FIAC). These inputs are either event reports notifying the FIAC of a fault condition or fault isolation commands issued from the site OCRI or from some other site.

Outputs:

Outputs from FIAC are issued by R(FIAC) and consist of solicited and unsolicited OCRI displays, measurement function module commands, and copies of event reports directed to other site FIAC modules. DBR requests are issued to the connectivity portion of the database to retrieve records related to event reports.

Row Components:

1) DC(FIAC):

This component serves to mediate all system input. The type of input (event report, command, or DBR result) is determined and a transfer buffer filled. Control is then passed to the appropriate component. Event reports cause control to be passed to DR(FIAC). Commands result in control passing to CCI(FIAC), and DBR results cause the DBR to be reentered.

2) DR(FIAC):

This component is entered when a event report is received by DC(FIAC). The current-fault table is examined for correlation with the event report. If a correlation cannot be immediately determined the DBR is called to retrieve the database connectivity information associated with the monitor point generating the event. The current-fault table is updated to reflect the current estimate of the fault location in terms of equipment connectivities. Control is then passed to DA(FIAC).

3) DA(FIAC):

This component examines the transfer buffer and current-fault table to determine which measurement function module commands to issue. If a determination can be made based upon the available information as to the probable source of the fault, a transfer buffer is prepared which will be dispatched by R(FIAC) to the site OCRI.

4) R(FIAC):

This component issues any commands which were prepared by DA(FIAC), forwards a copy of the event report if any to the indicated sites via SSCI, and dispatches displays to OCRI. R(FIAC) receives control from either DA(FIAC) or CCI(FIAC). Control is passed to DC(FIAC).

5) CCI(FIAC):

This component receives control from DC(FIAC) when a command is received from either the site OCRI or another site FIAC. The command results in a call to DBR to determine which monitor points must participate in the isolation request. Specific measurement function module commands are then placed in a transfer buffer and control is passed to R(FIAC) to issue the commands.

6) DBR:

The DBR Component is entered from DR(FIAC) and CCI(FIAC) to retrieve connectivity records from the database. At the completion of the request control is returned to the calling component by the next component control parameter.

3.10 SSCI - Station to Station Communications Interface:

This module performs the routing and interface functions necessary to permit communications traffic to flow between site systems. Traffic origination within the site and destined for a module at another site is addressed to the SSCI. Traffic arriving from other sites at the SSCI is then routed to a module within the site or to the SSCI of another site if the current site is intermediate to the destination.

Inputs:

SSCI receives inter-site traffic over an interface per site and one interface located within the site system. All traffic is handled by the CI module which performs the routine functions. The results of DBR requests are also accepted by CI.

Outputs:

All outputs from SSCI are accomplished by CI over the same interfaces used for the reception of traffic. DBR requests are output over the site system interface.

Row Components

1) CI:

This component includes the inter-site communications interface hardware. The major function of this component is the routing of traffic between sites. Messages arriving from another site with a destination module designator within the site of which the SSCI is a member are stripped of the site addressing codes and sent over the site system interface to the designed module. Traffic arriving at the SSCI from within the site system is modified to include the appropriate site addressing code and sent over the proper inter-site communications interface as determined by the CI routing table.

If an inter-site communications interface is found to be not functioning or has a high error rate an OCRI display is prepared and placed in a transfer buffer for dispatch by R(SSCI). Until the communications interface is determined to be functioning, traffic leaving the site over the failed interface is buffered on the site database by calls to DBR. Once the interface is determined to be functioning the buffered traffic is retrieved and transferred over the interface.

2) R(SSCI):

This component is called only to notify the OCRI that one of the inter-site communications interfaces has failed to function or exceeded the error rate threshold set at the time of site installation. Control is then returned to CI.

3) DBR:

The DBR is called to perform the buffering operations discussed under the section on the CI component. The next component control will indicate CI. While a DBR request is pending control resides with CI.

4. MICROPROCESSOR STUDY AND ANALYSIS

4.1 Introduction

This section compares a number of microprocessors from the standpoint of attributes most important to the requirements of MSCDM, selects two of these as candidates for the final choice and describes these two in considerable detail. The microprocessors under initial consideration were the Burroughs BDS, the Motorola 6800, the Intel 8080, the Data General microNOVA, the Fairchild F8, the General Instrument 8000, the Mostec 65(0/1)X, the National SC/MP, the RCA 1802, the Signetics 2650, the DEC LSI-11, the Intersil 6100, the National Semiconductor PACE, and the TI TMS-9900. Table 4-1 gives an evaluation chart for the above microprocessors.

The microprocessors selected as a result of the study were the DEC LSI-11 and the TI TMS-9900. The discussion that follows indicates the reasons for the choices, describes the two in some detail and shows a comparison of the two against a common benchmark test.

4.1.1 Desirable Features

Preliminary analysis of the MSCDM modules provided a list of desirable features for the microprocessor selected to implement the modules. These desirable features include:

i) Higher Level Language Available: A higher level language is desirable for using structured programming techniques for implementing the MSCDM software modules.

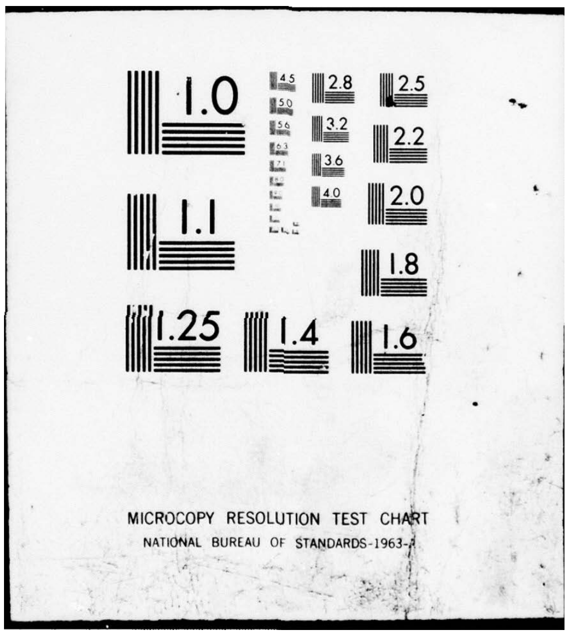
ii) Software Development Facility: A software development facility with editors, compilers, file handler and other utilities available for DCA personnel to do in-house software development after the Phase II equipment is delivered is desirable.

Table 4-1. Microprocessor Evaluation Chart

Micro-processor	Main Mfgr.	Language	Second Source	Development Tools	Cycle or Clock	Mini Compatibility	Opcode Set	Package	Code File Disk Space	Meets Req. Except	Unusal Feature
BDS	Burroughs	ALGOL	No	In-house	2 μsecs. per instruction	B 700 B 1700	COBOL-oriented micro-program	Six 51-pin + 3 ROM	Code plus file specifiers	Second source	Very small code files, bus. mach. compatibilities
6800	Motorola	PL/M	Yes	Exorciser	2-5 μsecs. per instruction	PDP-11	--	DIP	Many times code only	Code file size	
8080 (& Z-80)	Intel	PL/M	Many	ISIS-II	3 MHz. clock	--	Mini-like	DIP	Code plus some additional	Mini-comp. compatible	Most popular by far
microNOVA	Data General	FORTRAN	Yes	Yes	Fast	NOVA Eclipse	--	Several chips	Code plus some additional	Good hi-level lang.	
F8	Fairchild	--	License	Formulator	2 μsecs. per instruction	No	"Vertical" micro-program	1 chip "soon"	--	Several	"Vertical" microprogram
8000	General Instrument	FORTRAN	Europe	GIC 8000	500 kHz. clock	F8	Like F8	3 chips	--	Several	Older than F8
65 (0/1) x	MOStec	Soon	License	MDT	Up to 4 MHz. clock	Like 6800	Like 6800	DIP	--	Several	
SC/MP	Natl.	FORTRAN	No	LCDS	10-50 μsecs. per instruction	No	--	DIP	--	Several	
1802	RCA	No	License	COSMAC	2 1/2 μsec. per instruction 6.4 MHz. clock	No	--	DIP	--	Several	CMOS
2650	Signetics	FORTRAN	Yes	Board kits	10 μsecs. per instruction	None	Mini-like	1 chip	--	--	
LSI-11 MCP 1600	Western Digital	FORTRAN	DEC	DEC support for LSI-11	0.5 μsecs. per instruction	PDP-11	PDP-11	3 chips	Very large after binding	Code file size	PDP-11 very popular machine
6100	Intersil	PDP-8	Harris	PDP-8	5 μsecs. per instruction	PDP-8	PDP-8	1 chip	--	Several	PDP-8 may be obs
PACE	Natl.	No	No	Soon	10 μsecs. per instruction	NOVA	NOVA-like	1 chip	Code only?	--	
9(8/9)00	Texas Instruments	FORTRAN	No	Timeshare	3 MHz. clock	[illegible]	TI's own	DIP	--	--	I^2L

AD-A063 369 BURROUGHS CORP PAOLI PA FEDERAL AND SPECIAL SYSTEMS GROUP F/G 17/2
MODULAR SYSTEM CONTROL DEVELOPMENT MODEL (MSCDM). SECTIONS 1-7.(U)
SEP 77 DCA100-76-C-0083
UNCLASSIFIED 64280 SBIE-AD-E100 136 NL

2 OF 3
AD
A063369


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

iii) Down Line Loading Capability: A down line loading capability for loading the MSCDM processors via the selected distributed architecture is desirable.

iv) DMA or Block Transfer Capability: A direct memory access (DMA) or block transfer capability is desirable for providing high speed interprocessor communication via the communications architecture.

v) Word Size: An 8 or 16 bit word size is desirable to perform the MSCDM functions.

vi) Peripherals: A wide range of peripherals is desirable for the software development facility so that modular upgrades can be made as system requirements increase. For example, a station that currently uses mini-disk for data base storage can be upgraded to disk cartridge as data base storage requirements increase.

vii) Compatibility With Some More Powerful Computer: As system requirements increase it is desirable that the selected microprocessor is compatible with some more powerful computer (e.g., minicomputer, medium sized system).

viii) One-Board Microcomputer: It is desirable that the microprocessor selected be available as a one-board microcomputer which approaches the capability of a low-end minicomputer.

ix) Fast Multiply and Add Time: This speed requirement results from the VSQC module that uses a fast Fourier transform (FFT). Thus it is desirable that the microprocessor have a fast multiply and add time.

x) Low Cost and High Availability: A low cost and high availability is desired so that the Phase II equipment can be delivered on time and in a cost-effective manner.

4.1.2 The Selection Process

The microprocessor study began with a large number of microprocessors from which fourteen candidates (Table 4-1) were selected resulting in two acceptable candidates which satisfied all the desired features listed in Section 4.1.1. The availability of a suitable higher level language (i) and software development facility (ii) removed the Fairchild, Mostek, RCA, Signetics, Intersil, and PACE microprocessors from the group of final candidates. The development facility (ii) is desired to have acceptable peripherals (vi). Specifically, a mini-disk suitable for the storage of many programs and data was desirable, and this criteria removed the General Instrument and National microprocessors from further consideration. The fast multiply and add time requirement (ix) eliminated the Burroughs, Intel, and Motorola processors from further consideration. The low-cost and high availability requirement (x) eliminated the Data General MicroNOVA microprocessor from further consideration. The MicroNOVA processor is more expensive than the other two candidates, and the software development facility was less available with respect to both expected delivery date and accessibility for benchmark testing.

The remainder of this section describes the two candidate microprocessors (TI TMS-9900 and DEC LSI-11) in some detail. Both microprocessors are manufactured by recognized computer manufactures, have minicomputer compatibility, and can be programmed in FORTRAN IV. The TMS-9900 is faster, is a single chip processor, comes on a smaller microcomputer card, and is less expensive than the LSI-11. A language based on PASCAL is planned for the TI program development unit which will be available in late 1978.

A benchmark program written in FORTRAN was run on both microcomputers as well as their compatible minicomputers (TI990/10, PDP11/40). The benchmark program was translated into ALGOL and run on the B776 since it was readily available. Estimates for the current Burroughs BDS and near-future enhanced NBDS microprocessors were

then provided based on the B776 results even though the Burroughs entry was eliminated as a candidate.

One criterion which was not considered to be of major importance as a selection criterion was floating point capability. As mentioned above the speed requirements result primarily from the FFT calculation. Since the FFT used is a computationally stable algorithm with no significant cumulative truncation error, a 16 bit fixed point calculation achieves the required accuracy. Other modules where floating point could be used (e.g., trending analysis) were not considered to provide significant processor loads. Both the TMS 9900 and LSI-11 provide floating point capability if future requirements evolve with a desire for floating point. The LSI-11 requires the Extended Arithmetic Chip (KEV 11 option) for floating point; the TMS 9900 development system supplies floating point arithmetic routines as part of the runtime utility package supplied with the FORTRAN compiler. An Extended Arithmetic Chip for the TMS 9900 will be available in the very near future.

4.2 Microprocessor Systems Architectures

Differences exist in the packaging of the two types of processors, where the TMS9900 is a single chip (64 pin) unit, the LSI-11 is contained in 4 40 pin dual in line chips. Both utilize the concept of a bus to exchange data and information. The 64 pin package gives the TMS 9900 the luxury to provide separate memory data, memory address, I/O and control busses. The bus structure is shown in Figure 4-1. The LSI-11 shares the data and address bus for memory but provides for an additional 17 control lines, and the microprocessor chips communicate with each other over a 22 bit microinstruction bus. Separate data and address busses provide for a faster memory interface than a shared bus, eliminating the need for any multiplexing of address and data information. A typical LSI-11 configuration is shown in Figure 4-2.

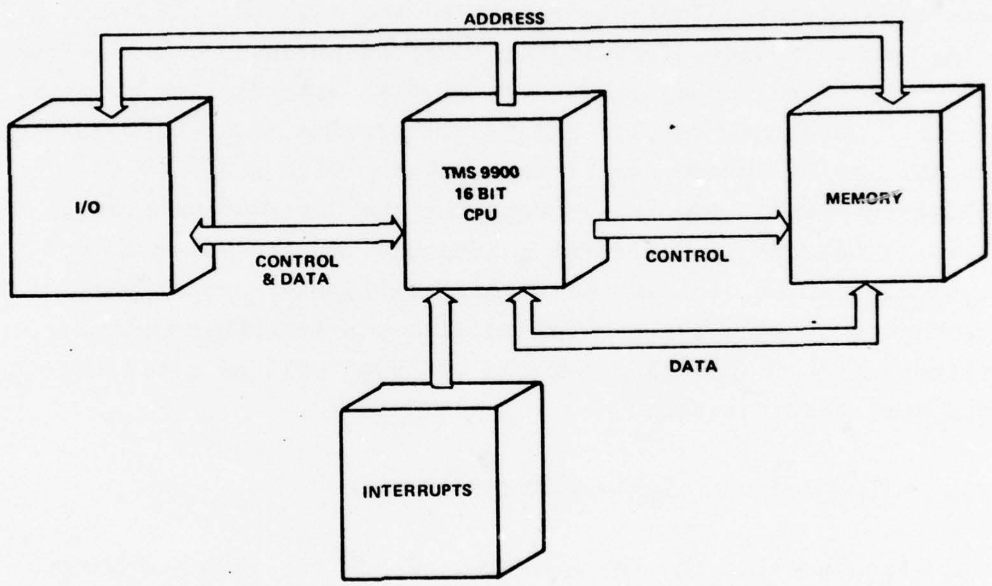


FIGURE 4-1. TMS 9900 SYSTEM BUS STRUCTURE

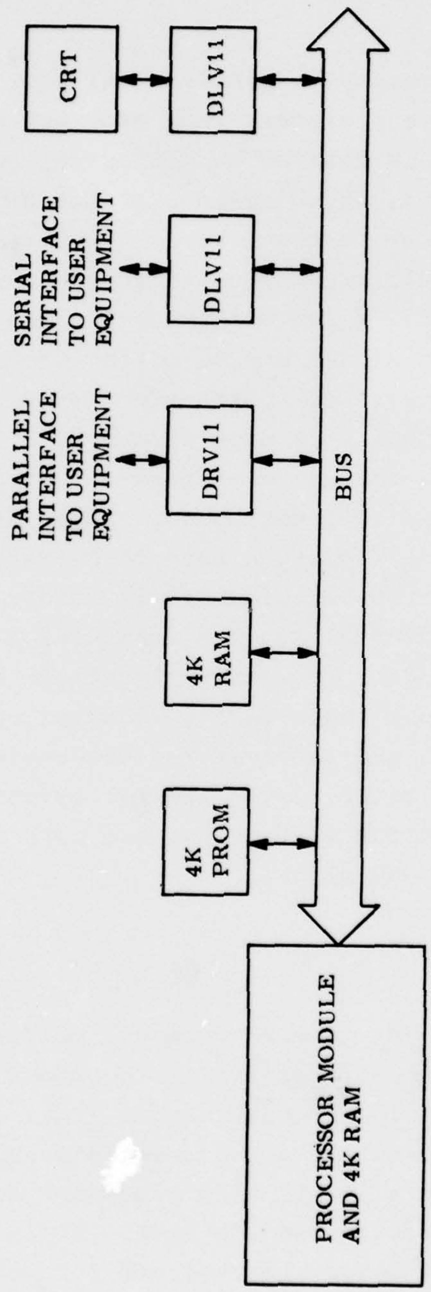


Figure 4-2
Typical LSI-11 Configuration

TMS 9900

The TMS 9900 utilizes a memory-to-memory architecture, which places the high usage data registers into external memory. A block of 16 memory words is defined as workspace, and located by the Workspace Pointer (WP), which resides on the CPU chip. WP contains the memory address of the first of 16 consecutive words in the workspace. Each different program routine, such as a subroutine call, can define a new workspace (Figure 4-3). Registers 13, 14 and 15 contain the WP, PC and ST of the previous routine and the Return Workspace Pointer instruction (RTWP) loads these values into these three registers upon return from this subroutine. The feature of locating workspace registers in external memory provides for fast response to interrupt or subroutine calls, since only the contents of three registers need to be saved. The processor interfaces with 16 bit wide memory words, where each word contains two bytes of 8 bits. The instruction set allows both word and byte operands. The memory locations are on even address boundaries. Memory space is 65,536 bytes or 32,768 words. The first 32 memory words are reserved for trap vectors (interrupts) and the second 32 memory words are for extended operation (XOP) instruction trap vectors. The last two memory words store the trap vector for the LOAD signal.

LSI-11

The processor contains eight general purpose registers for data storage, pointers, and accumulators. Two registers are dedicated to stack pointer (SP) and to program counter (PC). A hardware memory stack assists in handling structured data, subroutines, and interrupts. This is also a 16 bit wide processor and allows the same 64K byte addressing as in the TSM 9900. It is recommended that the last 4K bytes in memory be reserved for Peripheral and Device addresses. Device Interrupts and Trap Vectors are stored in the first 128 words of memory.

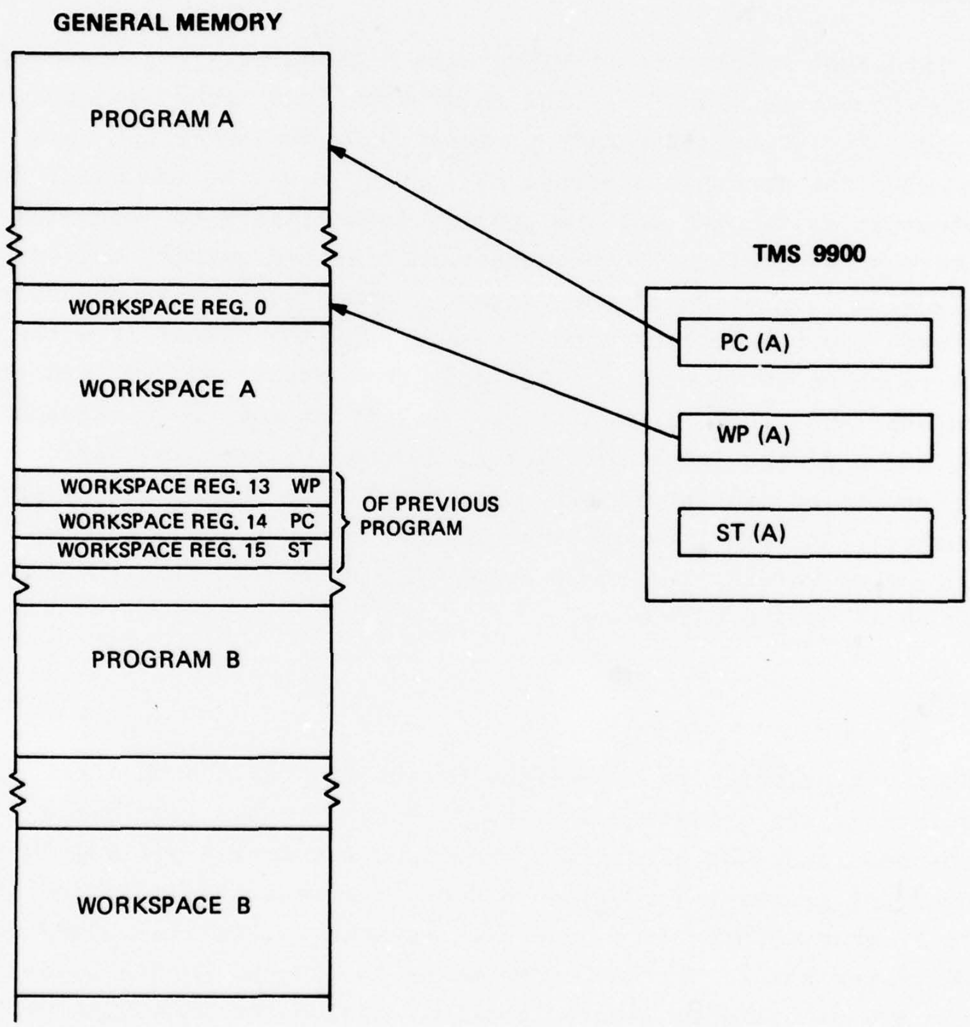


Figure 4-3

TMS 9900 Workspace Concept

4.2.1 Interrupts

TMS 9900

16 interrupt levels are provided with highest priority assigned to 0 and lowest to level 15. The input from four incoming interrupt control lines is continually compared with the interrupt mask and when the pending interrupt is less or equal to the mask, the interrupt is recognized. No polling is necessary to determine the interrupt's origin. The processor initiates a context switch following completion of the current instruction. The processor fetches the new workspace pointer and program counter from the interrupt vector location. The previous context WP, PC, and status register are stored in registers 13,14,15 of the new workspace. The value of the interrupt mask is reduced by one, so that only interrupts of higher priority will be able to interrupt the current routine. An (RTWP) return instruction terminates the service routine by reinserting the previous WP, PC and ST from locations 13, 14 and 15 of the workspace.

LSI-11

Interrupt priority is determined by position as the peripheral device interfaces are connected to the I/O bus. The bus provides a vectored interrupt eliminating the need for device polling during interrupt processing routines. Upon interrupt, the contents of the Program Counter (PC) and the Status Word (PS) are pushed onto the system stack. The new contents of the PC and PS are loaded from two preassigned memory locations called the interrupt vector. At the end of an interrupt routine the RTI (return from interrupt) pops the two top words of the stack and loads the PC and PS. Interrupt nesting is permitted and easily accomplished with the aid of the stack.

4.2.2 Microprogrammability

Both the LSI-11 and the TMS 9900 have a fixed instruction set (macro-instructions) and are not microprogrammable.

4.2.3 DMA or Block Transfer

Both microprocessors have a control signal which will suspend processor memory cycles, thereby giving other devices access to memory. DMA controllers for the LSI-11 are included on their single board computer package. Single chip DMA controllers for the TMS 9900 will be available next year.*

4.2.4 Memory Configuration

The TMS 9900 and the LSI-11 are both capable to address 32K words of memory (64K bytes). The only restrictions on the memory assignment are that the first few words at the beginning are reserved for interrupt vectors, etc. On single card prototypes, the first 4K of memory is usually included with the processor and additional memory space is obtained with off-board expansion memory modules. The selection of static, dynamic RAM and PROM or EPROM is strictly governed by the need of the system.

TMS 9900

The TM990/4 computer which is contained on a single card, contains the TSM9900 and provides for three types of memory to be mounted on the same module; i.e., EPROM, static RAM and dynamic RAM. On the card is room for up to 4K words of EPROM and 1/2K of static RAM. Off-board expansion of additional memory provides the full capacity of 32K. The TM 990/202 memory board (e.g.) has a capacity of up to 20K x 16 bits of static RAM, two such boards provide the total 32K word capacity. Depending upon the actual selection of the RAM chips the access time is between 150 and 450 nsec maximum.

---

*The unavailability of certain features in other microprocessors removed them from considerations. The presumed future availability of DMA allowed the TMS 9900 to remain in the running.

LSI-11

The KD11-F single board processor contains 4K dynamic RAM on the same module. Additional 4K boards with either dynamic RAM or core memories are available. Static RAM modules of 1K x 16 bits are also available. Access time is 550 nsec. Dynamic RAM modules of 16K x 16 bits are also available from DEC and from other memory manufacturers.

4.2.5 Peripherals

A wide variety of peripheral controllers exist for the candidate microprocessors, such as floppy disk, video display terminals, data communication interfaces, line printers, teletype terminals and tape cassettes. In the system under consideration, the individual microprocessors will have no peripherals other than the interface to the interprocessor communications network except for the program development unit which will interface with mini disk and a CRT.

4.2.6 Software Support

Software support for both processors is provided by operating systems which run on their respective language development systems. Compilers, assemblers, simulators, and link editors are available to assist in the language development phase of the project. Software development systems are described in Section 4.5. Crossassemblers between the 990 and the PDP-11 are available from T.I.

4.2.7 Documentation

Adequate documentation exists for both types of processors, since they are included in the product line by their manufacturers.

4.2.8 Design Aids

Single card processor modules such as the TM 990/xx and KD11-F are marketed with memory boards, interconnecting backplanes, power supplies and enclosures into development systems. Higher level language compilers will run on these systems if floppy disc storage systems are added as a peripheral. Description of the software development system is covered in Section 4.5.

4.2.9 Product Longevity

The two different microprocessors are components or parts in computing systems and are marketed as such. Improvements in the manufacturing technology will enhance the microprocessor's performance. The eventual goal is the replacement of the single board computer by a single chip.

4.2.10 Price and Availability

Copies of price lists are available from the vendors. Delivery times of 30 days have been promised after receipt of order. From OEM price lists in our possession, the TM 990 board is priced at $450 and the LSI-11 is priced at $990. LSI-11 compatible memories can be obtained at a price of approximately $1200 for 16K x 16 bits. TI memory can be obtained at a price of approximately $1400 for 20K x 16 and $900 for 10K x 16 bit modules.

4.3 Software Design Considerations

4.3.1 Higher Level Language Availability

<u>LSI-11</u>

DEC's operating system RT-11 supports the FORTRAN IV, BASIC, and FOCAL languages. RT-11 runs on an LSI-11 with between 8K and 28K

words of memory, and with the addition of a floppy disk drive allows one to compile programs written in FORTRAN IV or BASIC. RT-11 contains the following system utility programs:

- Text Editor
- Macro Assembler
- Macro Expander
- Assembler
- Linker
- Librarian
- On-Line Debugger
- Code Patch Utility
- Object Code Patch Utility
- Peripheral Interchange Program
- File Exchange Utility
- File Dump Utility
- Batch Processor

For the MSCDM application, the higher level language used will be FORTRAN IV. FORTRAN/RT-11 is an extended, optimizing FORTRAN IV compiler that processes source programs extremely rapidly. Typical 300-line programs compile in less than 25 seconds. The system is designed to minimize the size of executable programs. The RT-11 FORTRAN system subroutine library, SYSLIB, contains extensive string manipulation routines for creating strings in LOGICAL*1 arrays, and allowing their manipulation.

TMS 9900

The standard 990 software includes both memory-resident and disk-based operating systems. The programming languages encompass FORTRAN IV, COBOL, and Multi-user BASIC. Software development utilities are available to facilitate application program source editing, testing, assembling, and link editing. A floppy disk operating system for the 990/4 has recently been released and it

includes a FORTRAN IV compiler. The following features are included in the FORTRAN II compiler package:

- . Direct Access I/O
- . Overlapped I/O
- . Free Format Source Input
- . Literal Character Strings Represented in Quoted Form
- . Double Word (32-bit) Integer Data Types
- . An IMPLICIT statement to allow data type Declaration for Groups of Data
- . DATA Statement Array Names
- . Re-entrant Subprograms
- . Scaled Binary Data Types
- . Copy Directive
- . ACCEPT and DISPLAY directives for CRT interfacing.

In addition, a routine utility package performs the following services for the compiler:

- . Format Editing
- . ASCII To Binary and Binary to ASCII Converson
- . Floating Point Arithmetic Routines
- . FORTRAN Tracing

The DEC operating system is considered to be more mature than the TI operating system.

4.3.2 Addressing Modes

TMS 9900

Eight addressing modes are available:

- . Workspace Register Addressing R - Workspace register R contains operand.

- Workspace Register Indirect Addressing *R - Workspace register R contains the address of the operand.
- Workspace Register Indirect Auto Increment Addressing *R+ - Workspace register R contains the address of the operand. After operand fetch contents of R are incremented.
- Symbolic (Direct) Addressing at Label - The word following the instruction contains the address of the operand.
- Indexed Addressing at Table (R) - The word following the instruction contains the base address. Workspace register R contains the index value. The sum of base address and index value result in the effective address of the operand.
- Immediate Addressing - The word following the instruction contains the operand.
- Program Counter Relative Addressing - The 8 bit signed displacement in the right byte of the instruction is multiplied by 2 and added to the updated contents of the program counter. The result is placed in the PC.
- CRU Relative Addressing - The 8 bit signed displacement in the right byte of the instruction is added to the CRU base address of worskpace register 12. The result is the CRU address of the selected bit.

<u>LSI-11</u>

General Register Addressing:

R is a general register, 0 to 7
(R) is the contents of that register

- Mode 0 - Register - R-R contains the operand.
- Mode 1 - Register deferred - (R) - R contains the address.
- Mode 2 - Autoincrement - (R) + - R contains the address, then increment (R).

- Mode 3 - Autoincrement deferred - @ (R) + - R contains the address of address, then increment (R) by 2.
- Mode 4 - Autodecrement - -(R) - Decrement (R), then R contains address
- Mode 5 - Autodecrement deferred - @-(R) - Decrement (R) by 2, then R contains address of address.
- Mode 6 - Index - X(R)-R + X is address.
- Mode 7 - Index deferred - @ X(R) - (R) + X is address of address.

Program Counter Addressing:

Register = 7

- Mode 2 - Immediate - #n - Operand n follows the instruction.
- Mode 3 - Absolute - @#A - Address A follows the instruction.
- Mode 6 - Relative - A - PC+4+X is the address.
- Mode 7 - Relative deferred - @A - PC+4+X is the address of the address.

4.4 Hardware Design Considerations

4.4.1 Architecture

Figure 4-4 shows the architecture of the TMS 9900 which is a single chip microprocessor produced by N-channel silicon-gate MOS technology. The architecture of the LSI-11 is shown in Figure 4-5, which is the complete processor module. The main function is contained in the four microprocessor chips, which are a control chip, a data chip, and two microinstruction ROM chips.

4.4.2 Master Clocks

A master clock chip is available for the TMS 9900, the TIM 9904 Clock Driver. It requires an external crystal (48 MHz) and generates the four phases at 3 MHz for use with the processor. A similar clock pulse generator is located on the KD11-F LSI-11 microcomputer board, which supplies the four clock phases.

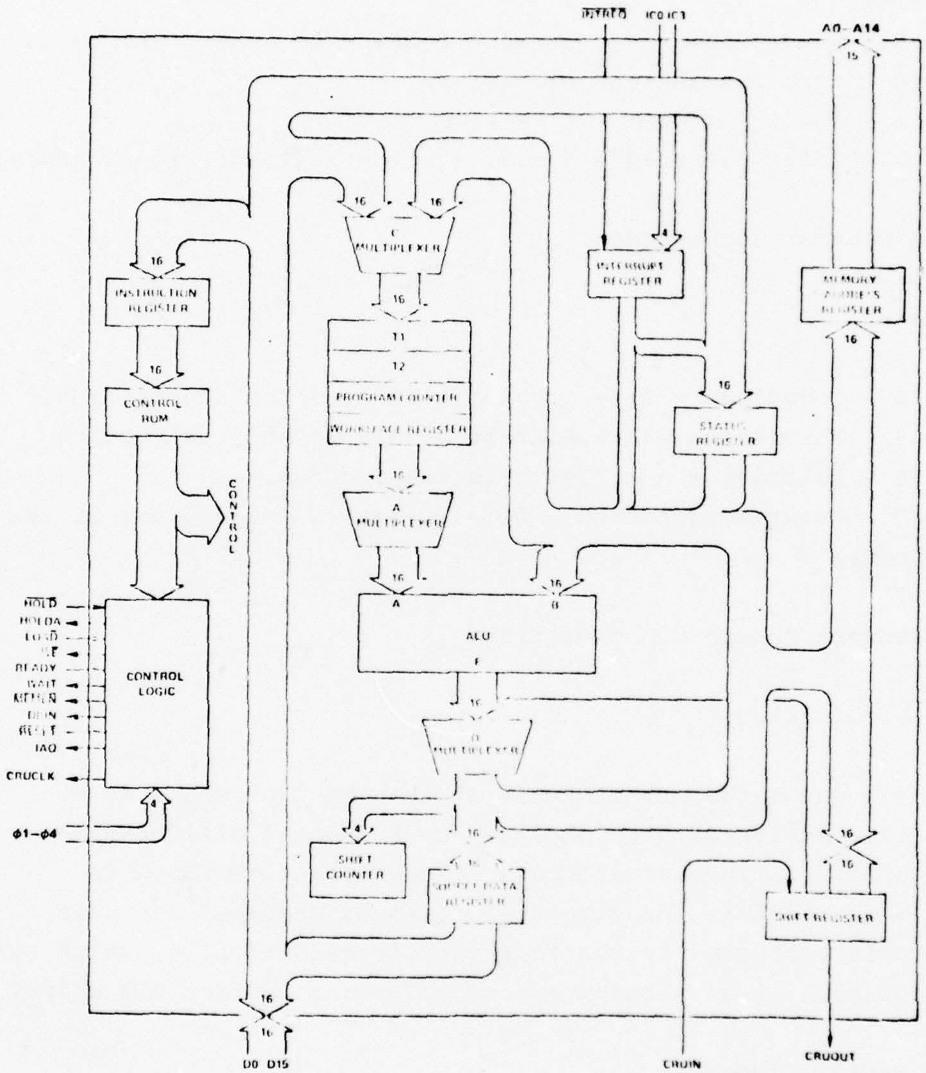


Figure 4-4 TMS 9900 Architecture

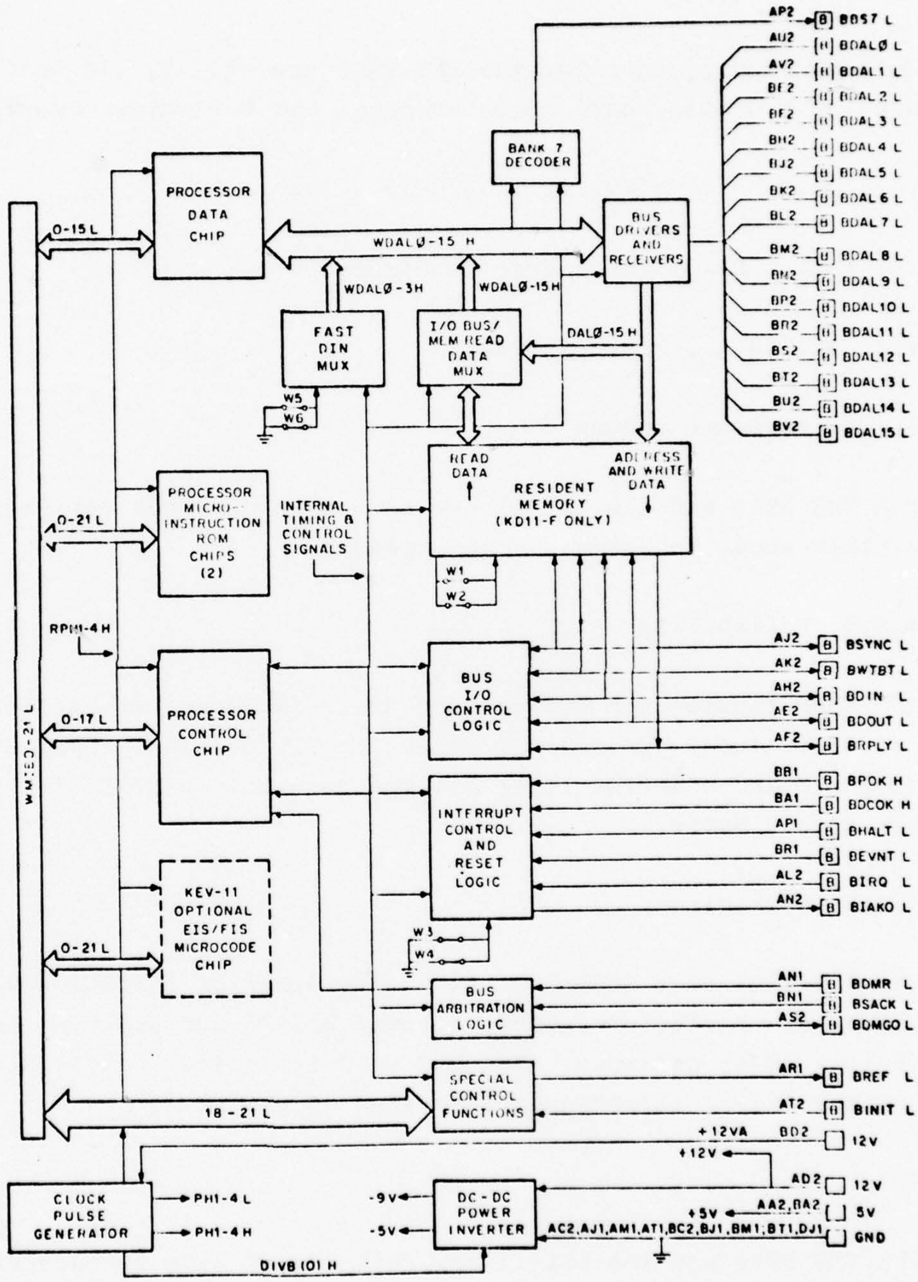


Figure 4-5 KD11-F Microcomputer Logic Block Diagram

4.4.3 Voltage and Power

Voltage requirements for the TMS 9900 are +5, -5, +12 volts. The 990/100 M single board computer needs the following power:

+5V. at 1.3 A., +12V. at .2 A, -12 V. at .1 A.

The KD11-F LSI-11 has these requirements:

+5V. at 1.8 A., +12V. at .8A.

4.4.4 Hardware Speeds

The TMS 9900 and the LSI-11 are both N channel MOS devices and exhibit about the same circuit speed.

4.4.5 Reliability

The MTBF figures for MOS devices are 2 failures per million hours and .5 failures per million hours for TTL devices. Applied to a single board computer these figures result in a MTBF of 1 failure per 25,000 hours.

4.4.6 Packaging

The semiconductor industry has standardized practically every aspect of manufacture, test, circuit boards and sockets for dual-in-line (DIL) packages. The TMS 9900 is housed in a 64 pin DIL and the LSI-11 is packaged into four 40 pin DILs.

4.4.7 Word Size

The TMS 9900 and the LSI-11 are both 16 bit wide processors with full 16 bit transfers to memory.

4.4.8 Address Capability

The addressing capability of the two microprocessors is 64K bytes, the maximum possible with a 16 bit wide memory address.

4.4.9 Execution Time

The TMS 9900 is generally faster than the LSI-11 by about a factor of 1.5. The execution time for I = J * K, for example, is about 32 us for the TMS 9900 and 65 us for the LSI-11. This is especially important in applications such as FFT's.

4.4.10 Registers

The TMS 9900 has 16 external registers located in each assigned workspace. In addition there are three registers accessible to the user, the Program Counter (PC), the Status Register (ST), and the Workspace Pointer (WP).

The LSI-11 contains eight internal registers where two are reserved for the Stack Pointer (SP) and Program Counter (PC).

4.4.11 Compatibility

The microprocessor circuits are all TTL compatible.

4.4.12 Environmental Effects

Operating temperature for these processors is 0°C to 70°C and storage temperature -55°C to 150°C.

4.4.13 Circuit Technology

The N-channel MOS technology is presently the most popular process for fabrication of microprocessor chips. They are still improving circuit complexity, circuit density, and circuit speed. The I^2L technology shows great future promise since it combines the

speed advantages of the bi-polar process with the circuit density of the MOS process. It has the additional features that only a single voltage is needed and that it shows improved resistance to radiation, which makes it attractive for military applications. An I^2L version of the TMS 9900 (the SBP9900) is available.

4.4.14 Maintainability

Since these processors are parts of computer systems such as the DEC PDP-11/03 and T.I. 990 systems, future maintainability is assured. Some manufacturers guarantee upward compatibility of their old circuits to the new improved circuits without the need for changes in the program.

4.5 Microprocessor Development Systems

A microprocessor Development System supports the programmer with the development, debugging and writing of programs for micro-computers. Mostly, the microprocessor is combined with a floppy disk and a visual display terminal into a computer system which operates under the control of an operating system program, allowing programs to be evaluated which are written either in machine assembly language or in some higher level language.

Model FS 990/4 Development System (Texas Instruments) includes a 990/4 single card microcomputer with 24K, 16 bit words of memory, floppy disk ROM loader, a video display terminal, model 913, and dual floppy disk units (model FD800).

It includes the TX 990/TXDS operating system with supporting development software utilities. Some features which are included are source editor, assembler and link editor for Fortran and AMPL, a Pascal type higher level language. All modules are housed in an enclosure which includes the power supply.

Model PDP-11V03 Development System (Digital Equipment) consists of a PDP-11/03 single board computer with 8K RAM memory, a visual display terminal (Model VT52 DECscope), and an enclosure cabinet with power supply.

The RT-11 operating system for model PDP-11V03 handles FORTRAN and BASIC as higher level languages, and in addition supplies a number of utility routines. A third higher level language FOCAL is also included in RT-11. The FORTRAN compiler and linker will run with a minimal 8K RAM memory system.

The approximate cost of the Development Systems is as follows.

Texas Instrument:

FS990/4 - CPU with 24K RAM, 512 ROM, Model 913 Video Display Terminal (12 line)and model FD800 Floppy Disk (484K bytes)	$10,500

Digital Equipment:

PDP11V03-AA - CPU with 8K RAM memory, dual drive floppy disk, (512K bytes) VT52 keyboard CRT, (24 line), cabinet and RT-11 software package	10,950
16K x 16 dynamic RAM (other Mfr, Monolithic Systems, Inc.)	1,200
	$12,150

Additional memory will be supplied in each case to bring the total memory to 32K x 16.

In addition to the above, in-circuit emulators may be added at additional cost to aid in debugging and maintenance of external processing modules.

A disk cartridge system capable of storing 7.5 M byte (2.5 M byte removable, 5M byte fixed) can be substituted for the floppy disk at a total development system cost of $24,500 for the DEC system.

For the TI system a 990/10 minicmputer can be used as the software development system. A 990/10 processor with 64K x 16 words of memory, dual 10 Mbyte disk cartridge, (5K fixed, 5K removable), CRT, and DX10 multitasking multiuser operating system sells for $27,400.

4.6 Benchmark Test

The program for the benchmark test was taken from a Purdue University publication by P. M. Lin et al.: "STARTUP - A STATISTICAL ANALYSIS AND ROUTING TABLE UPDATING PROGRAM FOR EUROPEAN AUTOVON", TR-EE 7622, Aug. 1976- Purdue University, West Lafayette, Indiana 47907. The timed element of this program was the sub-program PATH.

The program PATH finds, stores and writes the paths between all node pairs of a network. The first example of a 5 - node, 8 - link network was taken as the test data for the PATH program. PATH was written in FORTRAN and debugged on the PDP-11/40. It could be applied almost unaltered to the Fortran compilers of the LSI-11 and the TMS 9900 systems. PATH was rewritten into ALGOL so it could be compiled on the Burroughs "Stack" machine to generate machine code for the B776 and the BDS, for a wider comparison even though the BDS is not a final candidate.

Since the size of the RAM memories on the microprocessors were limited to 64K bytes some reduction in the arrays of PATH was necessary. The two arrays LOCATI and LOCATF were reduced from the original 15 by 15 by 5, to 5 by 15 by 5. The other array sizes were reduced accordingly and the format declarations needed some modifications.

Listed below are the running times for execution of PATH program looped 10 times

PDP-11/40	1.4 seconds
LSI-11	1.7 seconds
T.I. 990/10	0.5 seconds
T.I. 990/4	0.7 seconds
B776	6.6

Times for the BDS and nBDS were estimated to be 5 seconds and 2.5 seconds respectively based upon the B776 result. The rank-ordering by execution time was

TI 990/10	0.5 seconds	(Minicomputer)
TI 990/4	0.7 seconds	(Microcomputer)
DEC PDP-11/40	1.4 seconds	(Minicomputer)
DEC LSI-11	1.7 seconds	(Microcomputer)
Burroughs NBDS	2.5 seconds	(Microcomputer)
Burroughs BDS	5.0 seconds	(Microcomputer)
Burroughs B776	6.6 seconds	(Minicomputer)

For purposes of comparison, Appendix E gives the program description, the TI program compiled, the TI printout and results, the LSI-11 printout and results and the Burroughs B776 compilation, printout and results.

4.7 Conclusions and Recommendations

The TMS 9900 and the LSI-11 are both well-suited for the MSCDM application in that they satisfy all the desirable features listed in Section 4.1.1. Benchmark and instruction time comparisons indicate that the TMS 9900 will be approximately 1.5 to 2.5 times faster than the LSI-11 in the MSCDM application. Also the TMS 9900 processor is roughly half the price of the LSI-11 ($450 vs. $990). Memory costs for the two processors are approximately equal. The TMS 9900 processor is contained on a single chip and the LSI-11 processor is contained on 4 chips. The LSI-11 has the advantage of a longer existence with an older software development system. DMA capability is not currently available and an extra clock chip is required for the TI-990. However, based upon the fact that the TMS 9900 is cheaper and faster than the LSI-11, the recommended MSCDM microprocessor is the TI TMS 9900.

References:

(1) Texas Instruments 990 Computer Family Systems Handbook. Manual No. 945250-9701, Dec. 1975.

(2) Texas Instruments TMS 9900 Microprocessor Data Manual, Dec. 1776.

(3) Digital Equipment Corp. LSI-11, PDP-11/03 User's Manual, 1975.

(4) Digital Equipment Corp. Microcomputer Handbook, 1976.

(5) ATEC Software Conversion Study Final Report, R4968-1-1, Computer Sciences Corp., Dec. 1976.

5. DISTRIBUTED ARCHITECTURES

5.1 Introduction:

This section will address the subject of Distributed Architectures with particular reference to the use of micro-processors as the primary processing elements.

Because there is no universally agreed upon meaning for the term Distributed Architecture, a definition suitable for this study will be discussed. Several workers have addressed the definitional problem recently. In particular, (Jensen 76) has written:

> "In its most general sense, "Distributed Processing" entails a processing load partitioned across multiple processors which intercommunicate. This encompasses ... multiple general-purpose processors. The last of these appears to be of most general interest, and is often what is meant by the term "Distributed Processing."

(Jensen 76) is primarily concerned with the "Topological Structure" of the intercommunication between processors, and provides a taxonomy covering the various strategies which have either been employed or discussed in the literature. (Enslow 76) has proposed a more restrictive definition which is more along the lines of this study.

> "A distributed processing system must meet the following characteristics:
> 1) Two or more general purpose processors
> 2) A "System" operating system
> 3) The employment of a communications type protocol
> 4) Services are requested by name
> 5) Non-deterministic resource allocation.

He continues,

> "... by general-purpose processors, we imply a system in which processors are not dedicated or fully bound to specific tasks...".

In this application the Data Collection (DC) modules include special prupose hardware so that some of the processors must be bound to specific tasks; however, the other row modules may not necessarily be bound to the processing element of the DC modules. It can be seen that the above characteristics 3 through 5 follow from the characterization of modularity in Section 2 of this report. The use of a communications type protocol (i.e., message transfers) says nothing about the topological structure of the intercommunication between processors. A classical tightly coupled multiple processor system may employ a message transfer methodology amongst processes (e.g. Burroughs B6700/7700). It is recognized that the transfer delay between sender and receiver, and the costs of the interconnection paths (see (Anderson 75) are important factors in evaluating different architectures. With respect to this study, factors such as geographical dispersibility are not a primary concern since the inter-station connection toploogy is a separate issue (see the ESM and ESMD contracts). The primary purpose of this section is to examine several reasonable ways of achieving the intra-station connections between processes (i.e., column modules). Assuming a given collection of processing elements and associated memory and special hardware, the problem is to characterize the interconnection schemes in such a way as to facilitate the evaluation of proposed feasibility development model architectures.

5.2 Architecture Description

Six distributed architectures (Pluribus, CM*, Hierarchical Multi-Microprocessor, MINERVA, Ethernet, Loop) were investigated. The comparison of architectures assumes that the MSCDM column modules of Chapter 3 are mapped onto the microprocessor modules described in Section 4. Information on all architectures is obtained from the liaterature and references are provided. Additional informaton on the loop network is provided based on design and implementation experience unique to the Burroughs Corporation. A general tutorial on the loop architecture and example loop system descriptions built by Burroughs are provided in Appendix B.

The architectures are presented (Figures 5-1 through 5-6) using the PMS notation made popular by Bell and Newell. The legend for this notation is presented in Table 5-1.

Table 5-1. PSM Legend

P	Processor
M	Memory
S	Switch
K	Control
_	Bus
D	I/O Device

5.2.1 BBN/PLURIBUS:

The Pluribus (Heart 73) architecture was originally designed to serve as a switching node for the ARPA network. The designers state:

"The system achieves unusual modularity and reliability by making all processors equivalent, so that any processor may perform any system task; thus systems can be easily configured to meet the throughput requirements of a particular job. The scheme for interconnecting processors, memories, and I/O is also modular, permitting interconnection cost to vary smoothly with system size. There is no executive and each processor determines its own task allocation."

Figure 5-1 shows a PMS diagram of the prototype Pluribus described in (Heart 73). From the figure it can be seen that the processors are equal in that no processor is dedicated to a particular I/O device, so that as long as the I/O device is functioning and there is at least one processor functioning, the device can be serviced. The global or shared memory is used to provide the buffer space for inter-communications between devices and processors and to hold the code of programs to be executed. The memory colocated with the processors can be thought of as a combination program and data cache, in that it provides rapid access to private data and code without suffering the contention of accesses to the shared memory. The busses between processors and the shared memory are bit parallel and must be as wide as the address width plus the data width. In the case of the Sue processors this is a 32-bit wide bus. The bus requirements for the micro-processors under consideration would be 16 + 16 not including the necessary control signals to memory. Accesses to the shared memory are a word at a time, consequently the bandwidth of the bus must be matched to the execution rate of the processor in order to maintain effective utilization of the processor. The bandwidth of the bus is determined by the product of the bus width in bits and the bit rate for one bit. In this case the bit rate would be on the order of .25 - .5 MBPS, so that the bandwidth of the bus is on the order of 4 - 8 MBPS. The number of interconnecting busses (BTOTAL) in this architecture is dependent upon the number of processor busses (BP), the number of I/O busses (BI), and the number of shared memory busses (BM):

BTOTAL = BP*BI + BP*BM + BI*BM

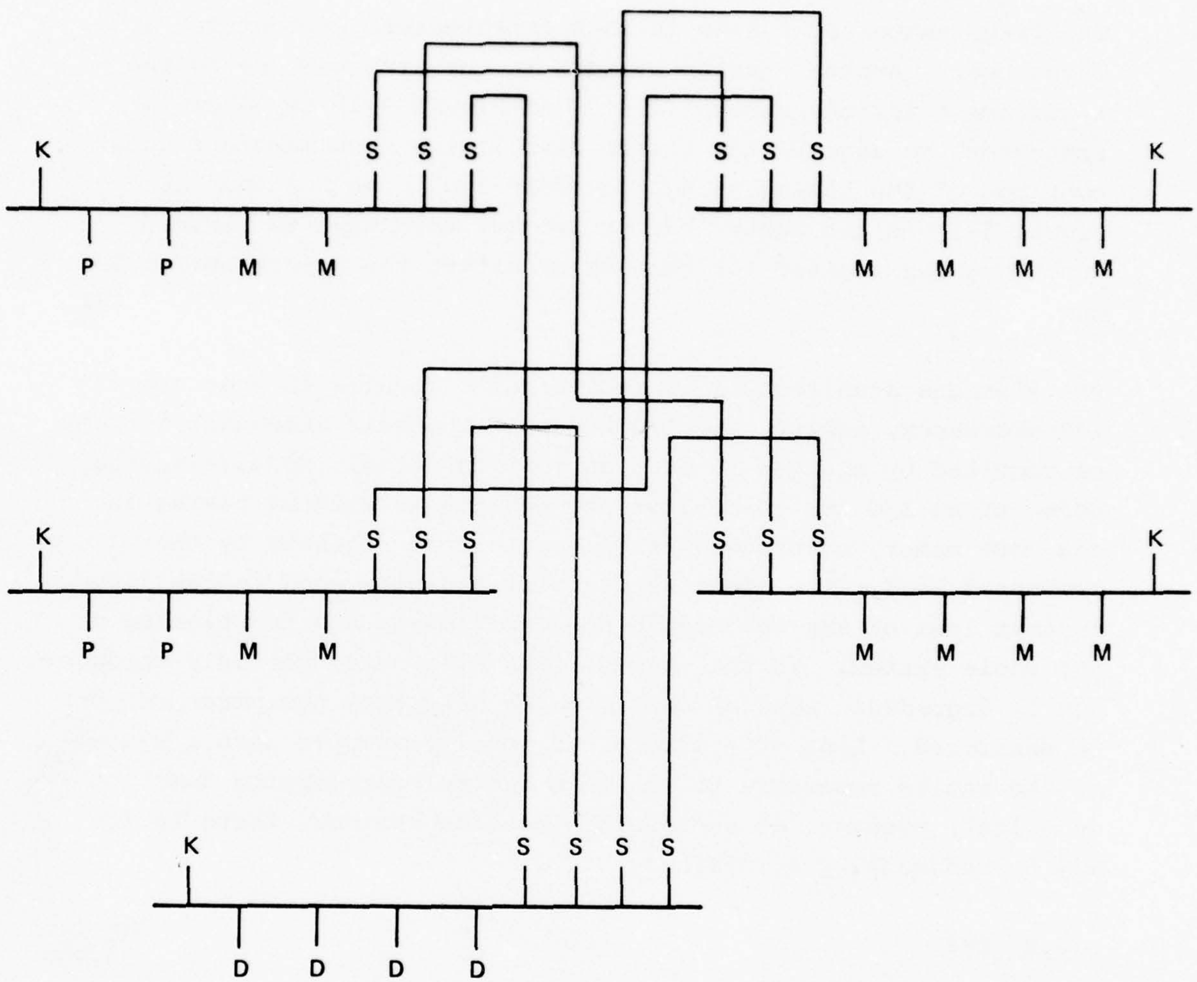


Figure 5-1. BBN/PLURIBUS PMS Diagram

The large number of busses in this architecture account for a large cost. Another factor of cost in the architecture is the requirement for bus controllers (K in Figure 5-1) to allow processors to acquire the bus so that it may then make a request over one of the busses going to memory I/O. The bus ends (S in Figure 5-1) on the memory bus or I/O bus then have to contend through a bus arbiter for the use of either the memory bus or I/O bus.

The Pluribus architecture is indeed quite modular in that one can add processor, memory, and I/O busses with their attendant devices as required by changes in station configuration. Modules may be added to an I/O bus to monitor new equipments without having to add more memory or processors unless this is required by the increased load. The architecture also exhibits good reliability in that loss of any one module does not impair the functioning of the whole system. In the case of loss of a processor only throughput is degraded. Loss of an I/O means only that equipment may not be monitored. Loss of a memory can totally cripple such a system if the tables necessary to maintain the multiprocessing function were lost; however, at increased execution overhead these tables may be redundantly maintained.

5.2.2 CM*

The CM* (Swan 77) architecture is a multiprocessor architecture intended to be built around microprocessors. It is an experimental system being developed at Carnegie Mellon University (CMU). Figure 5-2 shows the PMS diagram for the CM*. From the figure it is apparent that the major differences between the CM* and the pluribus are that the CM* has no shared memory, and the CM* has no separate I/O bus. The I/O devices are interfaced directly

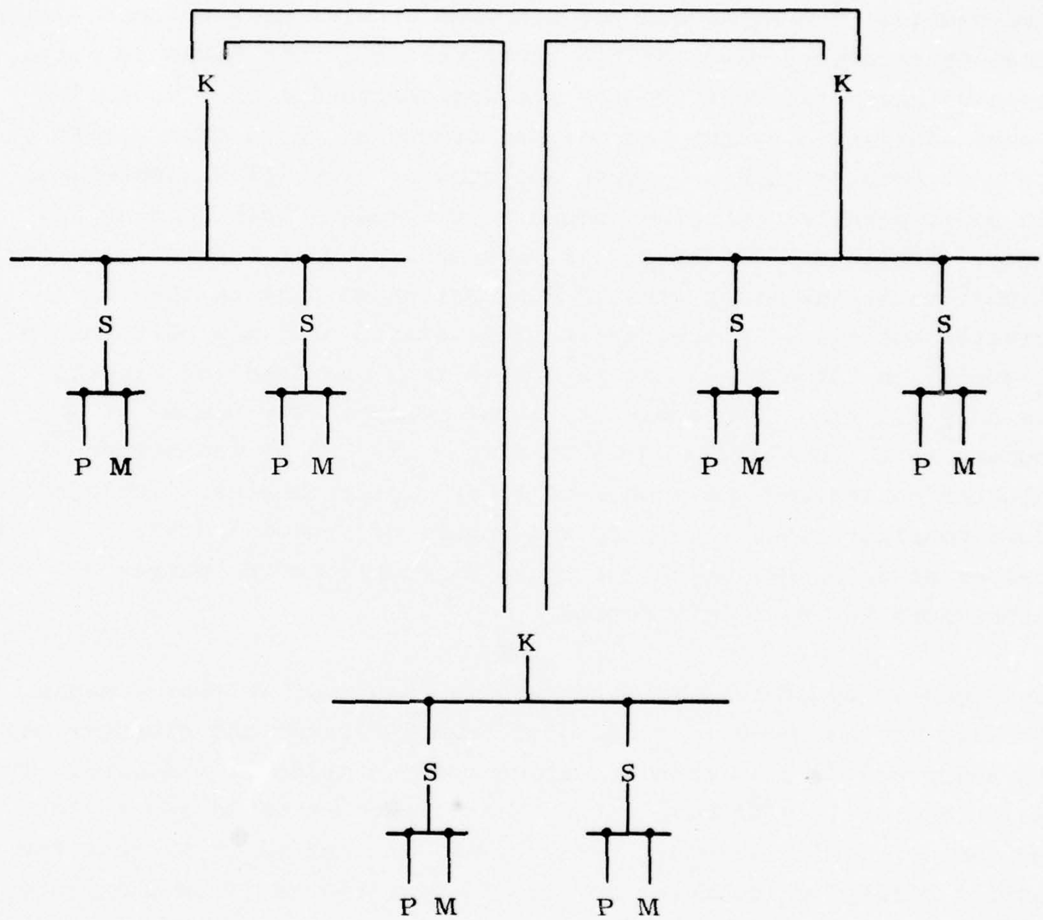


Figure 5-2. CM* PMS Diagram

to their respective computer modules; however, as long as the K.MAP and the S.LOCAL are functioning some other processor may service the device. The K.MAP and the S.LOCALS provide each processor with the entire memory space of the configuration. The K.MAP is quite sophisticated in providing for the implementation of a capability based addressing system, as well as providing "hard" management of data structures such as stacks and queues. The intra cluster bus is a bit parallel bus wide enough to accomodate both address and data. The inter-cluster bus is presumably also bit parallel. The bandwidth of the inter-cluster bus must be as high as the intra-cluster bus since in the report it is stated that the performance degradation for a non-local reference with one level of mapping is only 1.8 times the access to local memory. The number of busses in the CM* architecture is simply the sum of the number of intra-cluster busses and the number of inter-cluster busses. Included in cost considerations should be the number of K.MAPS and the number of S.LOCALS, which are equal in number to the number of processors in the configuration.

The modularity of this architecture is also good in that computer modules may be added to or deleted from a cluster and clusters may be added and deleted from a station configuration as required. The major reliability problem with the architecture would appear to be the critical placement of the K.MAP, in that if it is lost the entire cluster of computer modules is disconnected from the system.

5.2.3 SUNY/Hierarchical Multi-Microprocessor:

The hierarchical multiprocessor organization (Harris 77), attempts to strike a compromise between the time-shared single bus organization and the fully interconnected organizations such as the cross-bar switch type of interconnection strategy. By restricting the structure to essentially an "M"-way tree (Figure 5-3) some of the accessing generality is sacrificed for less cost in interconnections. In this architecture there is no arbitration problem for the busses

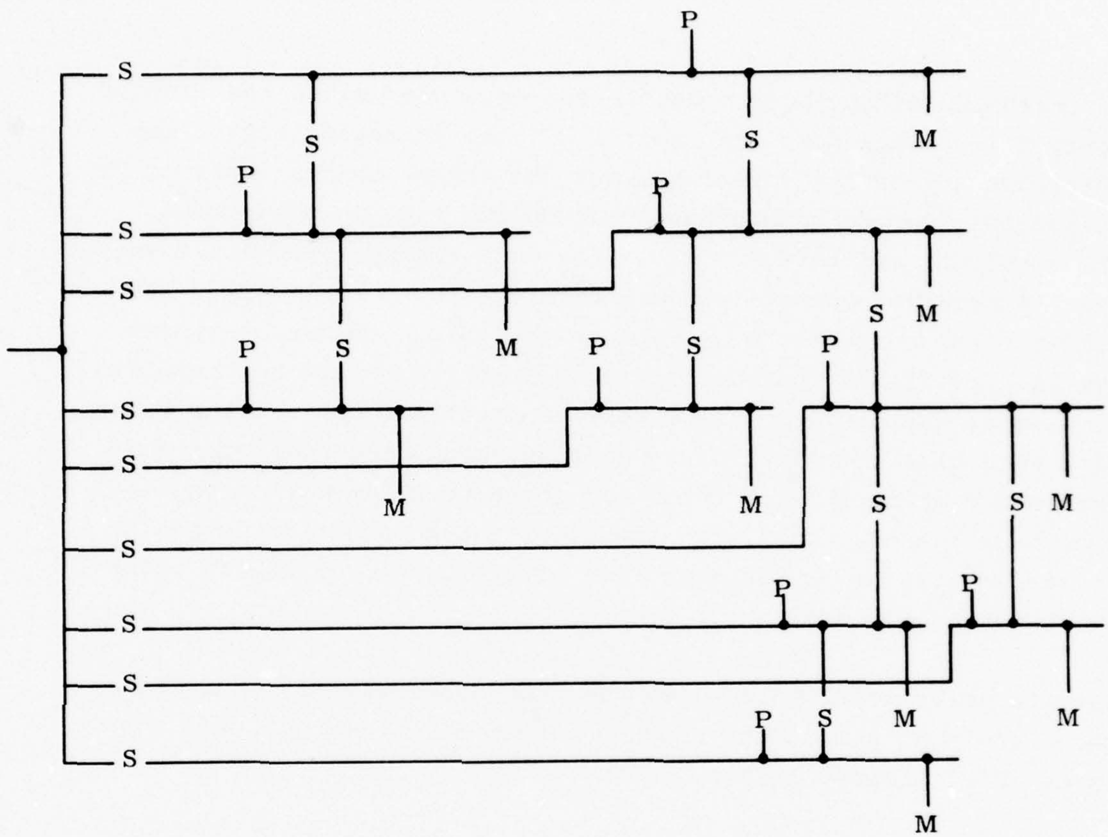


Figure 5-3 SUNY/HIERARCHICAL PMS Diagram

on which the offspring processors are connected since the line of control is always down the tree; i.e., the parent processor may interrupt any of its offspring, but the offspring must wait to be polled in order to communicate back up the tree to the parent. The structure off to the side is the data transfer mechanism whereas the hierarchical connections are reserved for control only. The hierarchical connections as well as the "side door" connections are bit parallel. The hierarchical connections must be the width of a word; whereas, the "side door" connections must be the width of a word plus the width of an address, since the side door connections must be able to fetch and store to the memories on the processor busses. Thus, the number of interconnection busses in an implementation is the number of offspring, BO, plus the total number of processors, BP.

As with previous architectures the interconnection costs include the S (Switch) components between parent and offspring, and the "side door" K(Control) components.

This architecture has less total processor flexibility than the previous architectures in that processors higher up the tree are dedicated to the control of processors lower in the tree. Further, this architecture uses a centralized control for inter-processor data transfers; thus as the number of processors grows this central data transfer control becomes a potential bottle-neck. There are two very weak points in the architecture from the reliability point-of-view. They are the centralized "side door" data transfer component and the root parent processor. If either of the two fails the entire system in incapacitated. Loss of any other processors simply disables a sub-tree of the system.

5.2.4 MINERVA/Multi-Microprocessor:

The Minerva (Widdoes 76) architecture is based on a single bit parallel bus with a centralized bus arbitration mechanism (see Figure 5-4). In this approach all processors, shared memories, and I/O devices are interfaced to a common bus. This allows all processors to be considered equal in the ability to service devices. Modularity is accomplished through the addition of ports to the bus arbitration device. This architecture can be seen to be essentially a simplification of the BBN/Pluribus architecture with less aggregate throughput. The number of busses in this architecture is one.

The complexity, and hence the cost, of the bus arbitration mechanism increases in direct proportion to the number of processors and I/O devices present in a configuration.

Unlike the BBN/Pluribus, the Minerva can allow only one conversation at a time. Furthermore, the centralized bus arbiter represents a critical component from the point-of-view of reliability. The aggregate throughput must be factored over all requestors so that as the number of requestors increases in a given configuration, the centralized bus arbiter can be expected to encounter the overhead involved in resolving conflicts arising from requests to use the bus.

5.2.5 XPARC/ETHERNET:

The XEROX ETHERNET (Metcalfe 75) was originally designed as an interconnection approach to be used to implement local networks. It is designed around a single bit serial bus. There are several important differences between the Ethernet architecture and the Minerva architecture. First, there is no shared memory, and secondly, the bus arbitration device has been de-centralized (see Figure 5-5). The bus interfaces (arbiters) in the Ethernet architecture are passive devices with respect to the bus. The bus is a coax cable into which a signal is injected by the bus interface. The bus interface is always receiving or monitoring the signals

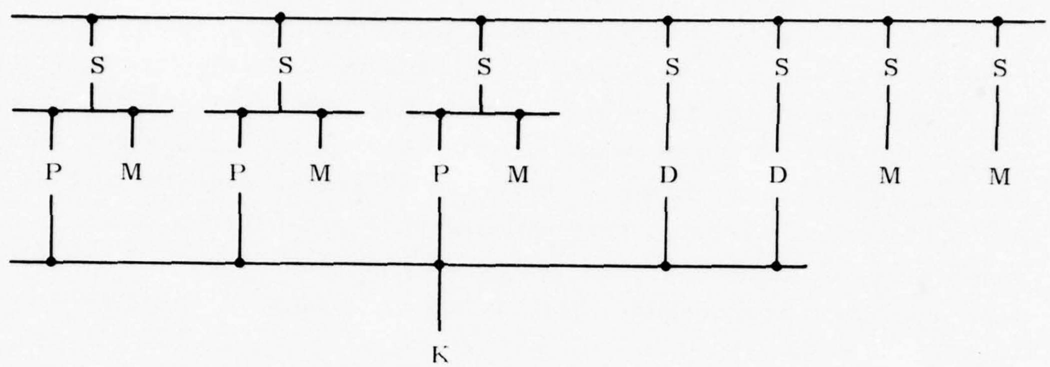


Figure 5-4. MINERVA PMS Diagram

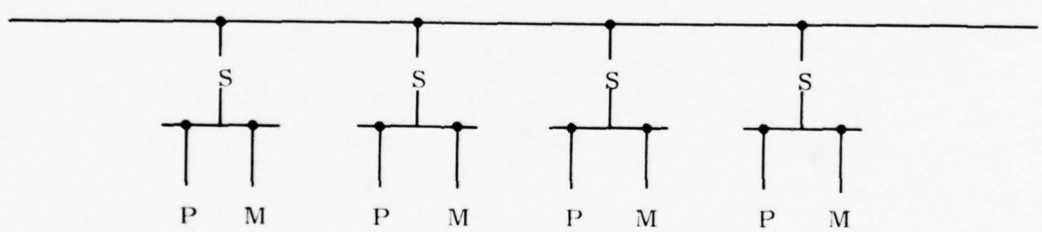


Figure 5-5. XPARC/ETHERNET PMS Diagram

on the bus. When a device requests the use of the bus, the interface determines if the bus is quiescent, which is done by detecting the presence or absence of a carrier which indicates that some other device has acquired the bus. If the bus is not in use, then the interface begins to transmit and monitor its transmission. If the interface reads a signal that is different from the one which it is transmitting then a collision has occurred and the transmission is aborted. Collision detection is necessary in this architecture because there is a finite delay in the transmission along the bus. Hence it is possible for two or more stations to simultaneously sense the bus as quiescent and begin transmission. Thus the Ethernet may be considered to use a statistical bus arbitration mechanism. When a collision is detected a backoff and retransmit algorithm is used to resolve contention for the bus. The bus is a single bit serial connection between all devices.

The cost of interconnection includes the bus interfaces which increase in number in direct proportion to the number of devices on the bus.

The modularity of this architecture is comparable to that of the previous architectures in that devices are simply added or deleted from the bus as necessary. The modularity is perhaps more meritorious in this approach than the Minerva, in that the bus interface is a single piece of hardware rather than an increasing more complex centralized mechanism. Further, because the bus arbitration is distributed there is no single component which can fail and disable the entire interconnection structure as it can in the Minerva architecture. Since the bus interfaces are passively coupled to the bus in most cases, failure of a bus interface will not disable the bus.

5.2.6 ADO/Loop:

The loop interconnection architecture as developed in ADO, Burroughs is primarily a single bit serial bus with a distributed bus arbitration mechanism as in the Ethernet (see Figure 5-6). Rather than being a broadcast bus with passive nodes, the loop may be viewed as an Ethernet bus with the ends connected, and active bus interfaces. That is, the bus interfaces are inserted in the signal path and must regenerate the signal as it passes around the bus. Another major difference is in the arbitration protocol. There are actually several protocols currently in use for loop type architectures (Farmer 69, Pierce 72, and Reames 76). The earliest approach (Farmer 69) uses the concept of a right to transmit which is passed around the bus from interface to interface. If a station wishes to acquire the bus it must wait for the 'write token' (WT) to be received by its bus interface before it can transmit, and must then pass the token to the next station on the bus after having transmitted. Two versions of this WT passing protocol exist; WT-1 transmits all packets at a node before sending the WT, WT-2 sends only one packet at a node before sending the WT. (Pierce 72) introduced the use of a circulating frame of fixed size packets which carried with each packet an inuse/available indication. With this approach, if a station wishes to transmit it waits until an available packet is detected, marks the packet inuse and then fills the packet with data. A receiving station marks the packet available and then removes the data from the packet. With this approach more than one station may be transmitting at a time. The approach of (Reames 76) attempts to merge the two previous approaches in that more than one station may be allowed to transmit (Pierce) variable length packets (Farmer). This is accomplished by implementing an elastic delay (queue) at each station's interface. A station wishing to acquire the bus must have enough available queue space to buffer any packets which arrive while transmitting. These conditions form the right to transmit which is the arbitration protocol.

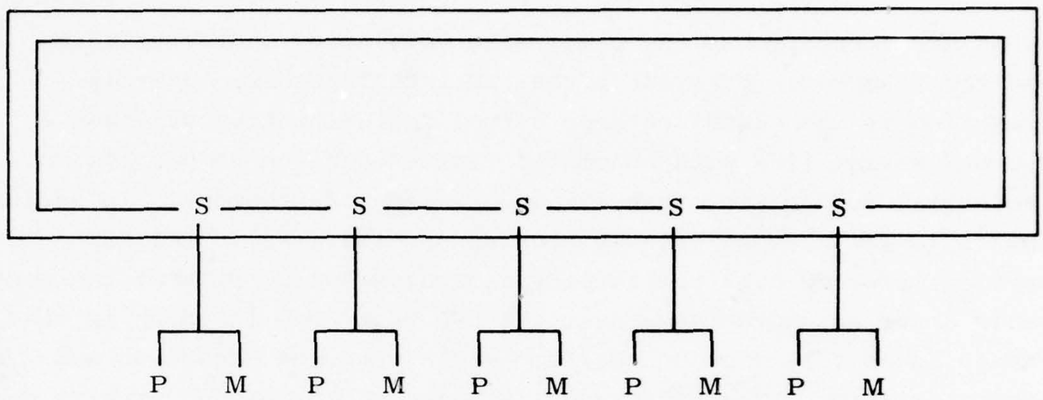


Figure 5-6. ADO/LOOP PMS Diagram

The interconnection costs include the bus interfaces which are directly proportional to the number of devices connected to the bus. Currently, ADO is studying implementations of all three protocols.

This architecture has the same basic modularity characteristics as the Ethernet. However, since the interfaces are actively inserted in the signal path an actual implementation utilizes a second serial line with automatic fault isolation to provide more potential reliability than the single line of Ethernet. In addition, multiple loops using gateway node connections (as in the ESM system) provide high reliability and throughput. Each of the protocols above has advantages over the Ethernet; one of which is that there is no collision on the loop which requires detection and retransmisison. The efficiency in terms of the amount of time during which there is useful traffic on the bus is 100% independent of loading; whereas, in the Ethernet efficiency drops as the bus becomes more and more loaded. Due to the statistical arbitration used in Ethernet there can be no guaranteed network acquisition time for a station; however, with the loop employing the Farmer protocol (WT-2) there is a guaranteed maximum network acquisition time. On the other hand, the Ethernet can support only one conversation at a time on the bus; however, using the Reames protocol there can be as many conversations as there are stations in the best case with the behavior of the loop asymptoting to that of the Farmer approach (i.e. one conversation). A detailed tutorial on loop communication networks including descriptions of example systems built by Burroughs is presented in Appendix B.

5.3 Architecture Comparison

This section evaluates the above six architectures in terms of a qualitative analysis of the sensitivity analysis criteria provided on pp. 2-69, 2-71 of the Technical Proposal. These criteria are: modular construction, throughput, simplicity of interface, relia-

bility, cost, software maintenance, and adaptability/flexibility. In addition, a criteria of low implementation risk is evaluated. The architectures are rated with respect to the criteria (excellent, good, fair, poor), and two candidate architectures are selected for the detailed sensitivity analysis of Chapter 6.

5.3.1 Modular Construction

Modular construction allows microprocessor nodes to be added and deleted easily as system requirements change. Inventory requirements and field servicing of equipment are greatly simplified. All the architectures surveyed exhibit modular construction. All candidates are rated excellent except for SUNY and MINERVA which are rated good. This is because in SUNY the central data transfer control becomes a potential bottle-neck as the number of processors grows; in MINERVA the central bus arbiter which resolves conflicts arising from requests to use the bus grows in complexity as the number of processors grows.

In addition to modular construction, one may evaluate architectural modularity which encompasses the ability to implement incremental changes in a system's capabilities. (Anderson 75) has identified two measures of modularity which are relevant to system architectures: 1) cost-modularity and 2) place-modularity. Cost-modularity refers to the cost incurred in adding an increment of system capacity, such as an additional processor or I/O device. Place-modularity refers to the restrictions placed on the choices of location and function for increments of system capacity.

The evaluation of cost-modularity is based on the increase in complexity of interconnection which is incurred in adding an additional element. BBN is considered fair since the addition of an element may require the addition of a new bus (Processor, I/O,

or memory). Adding a bus actually means adding a new set of interbus busses. CM* is rated good since the addition of an element may require the introduction of a K. map and two additional inter-cluster busses. The SUNY architecture is considered good since the addition of an element requires two busses (one for the side door and one to the parent). The remaining architectures are rated excellent since adding an element costs only the single element interface.

All architectures with the exception of SUNY are evaluated excellent with respect to place-modularity. This is because the architectures are in general insensitive to the placement of additional resources. The SUNY architecture however suffers from the throughput limitation of the side door data transfer mechanism and the restrictions on placement engendered by the control path structure.

5.3.2 Throughput

All architectures are rated excellent for throughput except MINERVA and ETHERNET which are rated good since they allow only one inter-processor communication at a time. However, the MSCDM throughput requirements are not very high. In Chapter 2 the absolute maximum burst rate identified was the 128 16 bit samples in 10 milliseconds which would be generated by the DC (VSQC) (assuming the DC is connected directly to the interprocessor communications network for reliability reasons rather than directly to a dedicated processor). This burst rate is equivalent to a 204.8K baud transfer rate. If it were necessary to be monitoring from three channels simultaneously in order to achieve the 1000 channels per hour required by the SOW, then the contribution to the maximum burst transfer rate required by the VSQC function would be 614.4 Kbaud. If we consider that the average data rate will be smaller than the above maximum burst rate and that all architectures may be assumed to employ transfer rates on single bit serial busses of 1-2 Mbaud, then the more connected architectures such as BBN/PLURIBUS are quite over-powered with respect to the MSCDM problem.

Table 5-2.
Architecture Evaluation for Modular Construction

BBN	excellent
CM*	excellent
SUNY	good
MINERVA	good
Ethernet	excellent
LOOP	excellent

Table 5-3.
Architecture Evaluation for Cost-Modularity

BBN	fair
CM*	good
SUNY	good
MINERVA	excellent
ETHERNET	excellent
LOOP	excellent

Table 5-4.
Architecture Evaluation for Place-Modularity

BBN	excellent
CM*	excellent
SUNY	fair
MINERVA	excellent
ETHERNET	excellent
LOOP	excellent

Table 5-5
Architecture Evaluation for Throughput

BBN	excellent
CM*	excellent
SUNY	excellent
MINERVA	good
ETHERNET	good
LOOP	excellent

Table 5-6
Architecture Evaluation for Simplicity of Interface

BBN	good
CM*	good
SUNY	good
MINERVA	good
ETHERNET	excellent
LOOP	excellent

5.3.3 Simplicity of Interface

Simple interfacing is required so that equipment (e.g., peripherals, data comm lines to other SYSCON elements, processors) can be easily added to the system as requirements evolve. The ETHERNET and LOOP were rated excellent; all other architectures were rated good. This is because ETHERNET and the loop are single bit serial bus architectures requiring very simple communication network interfaces as compared to the parallel bus architectures.

5.3.4 Reliability

Reliability (Survivability) implies that the system can operate in a degraded fashion even with selected nodal or equipment failures since control is distributed and there is no single system controller. All architectures were rated excellent with respect to reliability except for MINERVA (good) and SUNY (fair). MINERVA was rated only good because of the centralized arbiter. SUNY was rated fair since the entire system is incapacitated if the centralized "side door" data transfer component or the root parent processor should fail. In BBN/PLURIBUS, loss of a memory can totally cripple a system if the tables necessary to maintain the multiprocessing function were lost; to avoid this, these tables should be redundantly maintained. In CM*, a reliability problem appears to be the critical placement of the K.MAP in that if it is lost the entire cluster of computer modules is disconnected from the system. The loop architecture seems to have the best fault isolation capability of the architectures. Assuming the double loop architecture with automatic loop-back capability (cf. Appendix B), faults can be isolated such that the fault-causing node is removed from the loop. "Hot-card" replacement can then proceed without bringing the system down. As soon as the hardware is fixed, the loop reconfigures to include the previously-bad node. Additional reliability can be

achieved through multiple loops. A detailed discussion of loop reliability is given in Chapter 6.

Table 5-7
Architecture Evaluation for Reliability

BBN	Excellent
CM*	Excellent
SUNY	Fair
MINERVA	Good
ETHERNET	Excellent
LOOP	Excellent

5.3.5 Cost

If we assume that the module costs (i.e., microprocessors) are equal, for the different architectures then the relative system cost can be found by comparing the cost of implementing the interprocessor communication network. In general, high speed bit parallel busses are more expensive to implement than bit serial busses. This is because parallel busses require more expensive line drivers and receivers. The specialized bus controllers and bus arbiters for BBN/PLURIBUS add additional cost. The specialized parent-offspring processors, "side-door" connections, and root-parent processor add additional cost to the SUNY system. The K.MAPs and S.LOCALs add additional cost to CM*. The centralized bus arbiter adds additional cost to the MINERVA system. Based upon the above the single bit serial busses (ETHERNET, LOOP) were rated excellent, MINERVA was rated good, CM* was rated fair, and BBN and SUNY were rated poor.

Table 5-8
Architecture Evaluation for Cost

BBN	Poor
CM*	Fair
SUNY	Poor
MINERVA	Good
ETHERNET	Excellent
LOOP	Excellent

5.3.6 Software Maintenance

The organization of the communications control processing functions into separable and distinct microcomputer hardware/software modules that communicate through simple, generalized interfaces serve to simplify software development, verification, maintenance, and modification. In general, small simple software modules are easier to understand and maintain. The interprocessor communication software becomes more complicated and thus less maintainable as bus control arbiters are utilized and memory is shared. Based on the above the distributed control architectures of ETHERNET and LOOP were rated excellent and all others were rated good.

Table 5-9
Architecture Evaluation for Software Maintenance

BBN	Good
CM*	Good
SUNY	Good
MINERVA	Good
ETHERNET	Excellent
LOOP	Excellent

5.3.7 Adaptability/Flexibility

The MSCDM application requires a large degree of adaptability/ flexibility. For example, raw data can be input directly to the interprocessor communication network and routed to an appropriate microprocessor module. The processing load can be shared among a group of microprocessors, programs can be loaded into different microprocessors as MSCDM requirements change, spare modules can be loaded when other modules fail, and new microprocessors and peripherals can be interfaced to the network as technology evolves. All architectures have been rated as excellent with respect to adaptability/flexibility except SUNY which was rated as fair since processors higher up in the tree are dedicated to the control of processors lower in the tree and central data transfer control becomes a potential bottle-neck as the system is expanded. The loop network is extremely flexible in that different loop protocols can be implemented (e.g. WT-1, WT-2, Pierce, Reames). Burroughs is currently implementing a loop interface utilizing different control protocols (selectable depending on application and/or system environment).

Table 5-10
Architecture Evaluation for Adaptability/Flexibility

BBN	Excellent
CM*	Excellent
SUNY	Fair
MINERVA	Excellent
ETHERNET	Excellent
LOOP	Excellent

5.3.8 Low Implementation Risk

The low implementation risk criterion is included to measure the degree of development required to implement an architecture. This Phase I MSCDM report is used to recommend a Feasibility Development Model (FDM) which must be built and delivered as a turn-key system in Phase II. Thus it is imperative that the Phase I recommendation reflects an architecture that can be developed, built and delivered in a cost-effective and timely manner (e.g., 8 months, $37,626 cost). For this criterion the LOOP was rated as excellent. This is because much of the hardware/software/firmware is off-the-shelf in that it already has been developed at ADO, and Burroughs has considerable experience in implementing LOOP networks (cf. Appendix B). ETHERNET is rated as good since it uses a relatively simple interface to a bit serial bus. However, since the bus interface would have to be developed, it is estimated that this task would take 3-4 months more time and 6-8 man-months more development effort than the LOOP implementation. MINERVA was rated as fair since a parallel bus interface and bus arbiter development estimate would be 6-8 months more time and 12-16 man-months more development effort than the LOOP implementation. BBN, CM*, and SUNY were rated as poor since the development time and effort would be estimated to be similar to a medium to large scale computer system implementation (e.g., B900) due to the large number of busses and bus arbiters.

Table 5-11
Architecture Evaluation for Low Implementation Risk

BBN	Poor
CM*	Poor
SUNY	Poor
MINERVA	Fair
ETHERNET	Good
LOOP	Excellent

5.4 Conclusions and Recommendations

Based upon the above analysis, the MSCDM recommended architectures are the bit serial bus architectures, the ETHERNET and LOOP. Although the bit parallel bus architectures (BBN, CM*, SUNY, MINERVA) have sufficient processing power for the MSCDM application, they represent a considerable overkill and are more costly and have more implementation risk. The ETHERNET and LOOP are easier to implement, less expensive, easier to interface and maintain, have sufficient throughput, and high reliability.

A detailed comparison between the ETHERNET and LOOP will be given in Chapter 6 using specific Feasibility Development Model (FDM) designs consisting of the microprocessor module candidates (TMS9900, LSI-11) described in Chapter 4. As a result of this analysis a recommended FDM design will be presented.

REFERENCES

(Jensen 76) E. D. Jensen, et. al; A Review of Systematic Methods in Distributed Processor Interconnection. Proceedings ICC - 76 p. 7-17.

(Enslow 76) P. Enslow, Distributed Processing Workshop Transcript. Computer Architecture News 5:5 p.11

(Metcalfe 75) R. M. Metcalfe, D. R. Boggs; ETHERNET: Distributed Packet Switching For Local Computer Networks Xerox Palo Alto Research #CSL75-7

(Bell 71) G. Bell, A. Newell; Computer Structures and Examples Prentice Hall

(Heart 73) F. E. Heart, et. al.; A New Minicomputer/Multiprocessor for the ARPA Network AFIPS Vol. 42

(Swan 77) R. J. Swan, et. al.; The Structure and Architecture of CM*: A Modular, Multi-microprocessor Comp. Sc. Res. Rev 75/76 CMU p. 25

(Harris 77) J. A. Harris, D. R. Smith; Hierarchical Multiprocessor Organization Symposium on Computer Architecture #4 p. 41

(Widdoes 76) L. Widdoes; Minerva Multi-Microprocessor Symposium on Computer Architecture #3

(Farmer 69) W. D. Farmer and E. E. Newhall, "An Experimental Distributed Switching System to Handle Bursting Computer Traffic", Proc. ACM Symp. 'Problems in the Optimization of Data Communications Systems', Pine Mtn, Georgia (Oct. 1969)

(Pierce 72) J. R.Pierce, "Network for Block Switching of Data", BSTJ 51, 6, (July - Aug. 1972)

(Reames 76) C. B. Reames and M. T. Liu, "Design and Simulation of Distributed Loop Computer Network (DLCN)", Proc. of 3rd Annual Symp. of Comp. Architecture, Jan. 1976.

(Ramamoorthy 76) C. V. Ramamoorthy and T. Krishnarad, "The Design Issues in Distributed Computer Systems", Univ. of Cal. at Berkeley, 1976.

(Anderson 75) G. A. Anderson, E. D. Jensen; Computer Interconnection Structures: Taxonomy, Characteristics and Examples <u>Computing Surveys</u> Volume 7, Number 4, Dec. 1975.

6. CANDIDATE ARCHITECTURE ANALYSIS AND DESIGN

6.1 Introduction

This chapter provides detailed designs for the FDM simulation system implementation. The designs are compared, and a recommended design is chosen with implementation options provided for Phase II. The goal is to provide DCA with a powerful simulation facility implemented in a cost-effective and timely manner.

There are four permutations of designs discussed below. These result from two different microcomputer modules; namely the TI TMS 9900 and the DEC LSI-11. These two microcomputers are used in two candidate architectures; namely: the loop architecture, and the serial bus architecture exemplified by ETHERNET. These architectures are discussed in Chapter 5.

The candidate microcomputers are discussed in Chapter 4 which includes the results of a benchmark. Since the TMS 9900 seems considerably faster than the LSI-11 (by a factor of 1.5 to 2.5), the difference in speed had to be assessed in terms of a possible difference in the number of microcomputers required. The main requirement for speed is in the VSQC module. It is estimated that the LSI-11 module can perform quality control on 360 voice channels per hour whereas the TMS 9900 can perform the same function on 550 channels per hour. It is thus estimated that for 1000 channels of voice, three LSI-11's or two TMS 9900's would be required. This would mean that in the entire system, one more LSI-11 would be required over the TMS 9900 requirement.

It appears advantageous as a strategy of design that all microcomputer modules be identical to all other microcomputer modules in any particular system so that physical interchangability can be provided. The exception to this would be the program develop-

ment unit which would have dual minidisk units and a CRT input-output device. In the feasibility development model, the program development unit could also be used as the operator's console and local data base.

Regarding the architectures, all architectures discussed in Chapter 6 are adequate for the task involved. The parallel bus with arbiter architectures are considered to be too complex, too expensive and represent a considerable overkill. Of the two chosen for comparison, the loop with a one megabit/second capability and the serial buss (ETHERNET) with a similar capability are more than sufficient for the requirement.

In general, the communications architecture should be such that all of the necessary communications between microcomputers, the simulated inputs and the ESM be performed in an accurate and expeditious fashion without undue queuing of messages. By the same token, the communications architecture should not be uneconomic in terms of high speed for its own sake. The communications interfaces should be such that a good fit exists between the computer modules and the architecture that supports them. The microcomputer modules should be fast, accurate and have a memory complement that fits the requirement. In all of the above, the modules formed should be as much like other modules physically as is effective for the concept of interchangability without undue expense.

Software should be such that all machines can be operated using the same code repertoire generated from the same program development unit. It is advantageous that the progra- development unit be a part of the feasibility development model. The software should contain a high level language compiler for the development of module programs. In addition, suitable programming aids such as a disk file manager and a text editor should be available. The operating system should be a minidisk or disk cartridge operating system.

Microcomputer firmware should exist for program loading to the modules from the program development unit via the communications structure. In addition, microcomputer input/output firmware should be available as required.

The column functions described in Chapters 2 and 3 are listed below:

1. VSQC - Voice Service Quality Control
2. DSQC - Digital Service Quality Control
3. BBSA - Baseband Signal Analysis
4. WBSA - Wideband Signal Analysis
5. SDCA - Switch Data Collection/Analysis
6. OCRI - Operator Control and Report Interface
7. FIAC - Fault Isolation and Analysis Coordination
8. SSCI - Station to Station Communications Interface
9. DBMS - Data Base Management Service

These functions wil be mapped to specific microprocessor architectures. Interprocessor communication will be via the communication network architecture (e.g., loop, serial bus).

A Life Cycle Costing analysis is presented. Six alternative architectures are compared (LOOP-TI, LOOP-DEC, SERIAL-TI, SERIAL-DEC, PARALLEL-TI, PARALLEL-DEC). The PARALLEL-TI and PARALLEL-DEC architectures are not candidate architectures. These architectures were included in the Life Cycle Costing analysis for additional comparison, and to justify the conclusions of the qualitative architecture analysis of Chapter 5.

A simulation study was performed comparing the loop and bus architectures. The simulation was performed using the Burroughs Operational Systems Simulator (BOSS) on a B6700. The architectures were compared for a ten node system assuming equal Poisson inputs at each node.

6.2 Modules Mapped to a Loop Architecture

A typical mapping of column functions to hardware modules for an FDM connected in a loop configuration is shown in Figure 6-1. The functions and modules are interchangeable except for those modules that interface to specific hardware (e.g., DBMS connects to a mini-disk). The column function inputs and outputs defined in Chapter 3 are implemented via the loop. Different configurations for performing simulations can be accomplished by loading different column functions via the down-line loading capability that the loop supplies. A total of seven TMS 9900 modules or eight LSI-11 modules are used. Hardware requirements for the loop FDM using the TMS 9900 are given in Table 6-1. Hardware requirements for the loop FDM using the LSI-11 are given in Table 6-2. The Loop Interface Units (LIU's) are off-the-shelf hardware modules developed at Burroughs Advanced Development Organization.

The modules in Figure 6-1 would have the following functions:

1. Scenario input simulation: The B776 of ESMD loop 4 will generate input data to the FDM and provide the SSCI function. LIU #1 will connect to the BDS residing in the ESMD loop cabinet. In an actual system, real inputs would be interfaced to the loop.

2. SSCI: This LIU also connects to the BDS residing in ESMD loop 4. It would be used to provide inputs to other stations simulated by other ESM equipment.

3,4. VSQC, DSQC: These microcomputers operate on data received from the loop to provide the VSQC and DSQC functions. Two of these modules are required for the TMS 9900 microprocessor or three for the LSI-11.

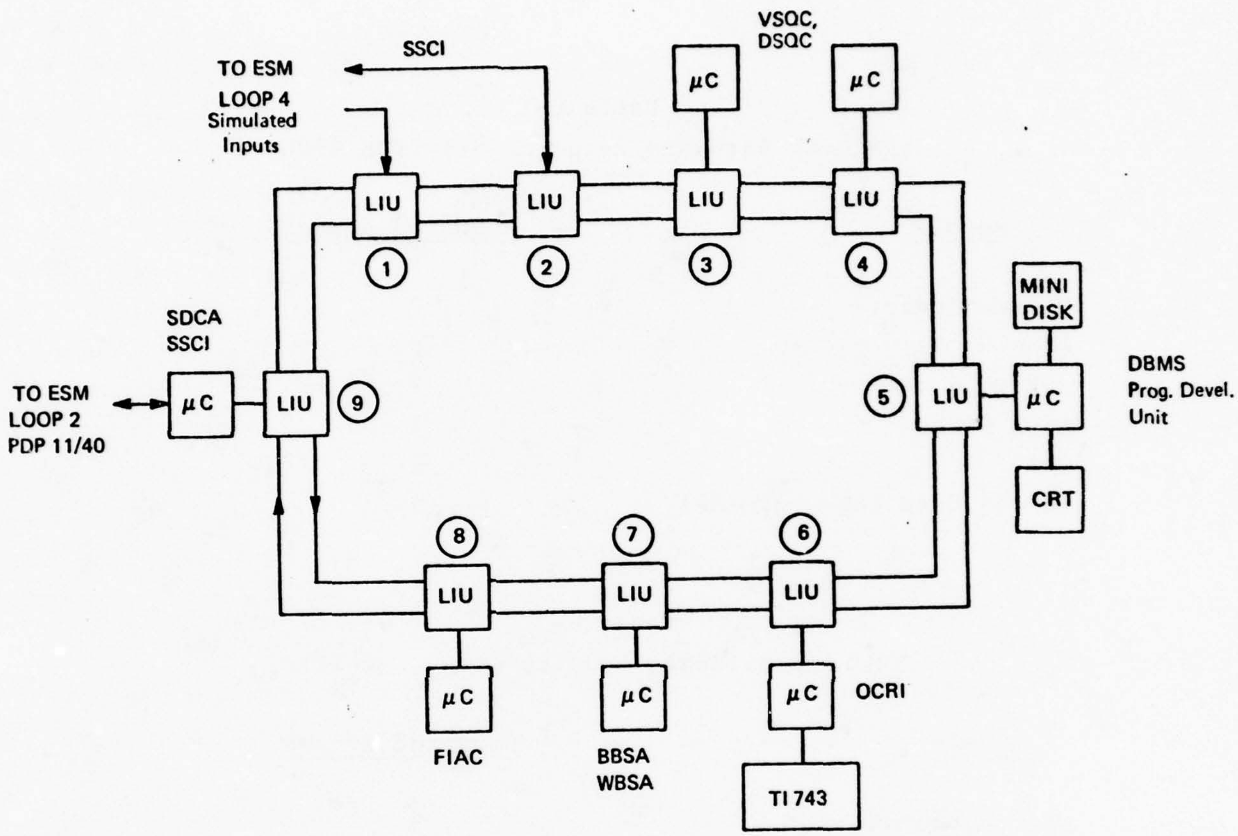


Figure 6-1

FDM Using Loop Architecture

Table 6-1

FDM Loop Hardware Requirements (TMS 9900)

Module	Required Number
Microcomputer	7
LIU	9
32K word memory	7
Mini Disk	1
CRT	1
TI 743 Hard Copy Terminal	1

Table 6-2

FDM Loop Hardware Requirements (LSI-11)

Module	Required Number
Microcomputer	8
LIU	10
32K word memory	8
Mini Disk	1
CRT	1
TI 743 Hard Copy Terminal	1

5. DBMS: The program development unit provides the DBMS function during simulations. If a minicomputer option is selected the DBMS and OCRI function can both be provided. This node also provides the system loading.

6. OCRI: This node performs the OCRI function. The implementation of the operator interface will be decided in Phase II. Alternatives include: i.) Let OCRI talk to the CRT terminal connected to the program development unit (PDU) via the loop. ii.) Install a switch on the CRT so that it can connect either to the PDU or the OCRI processor. iii.) Either supply or use an existing hard copy terminal (e.g., TI Silent 700) or CRT connected directly to the OCRI processor. iv.) Use a remote hard copy terminal (e.g., LA36 DECWRITER) on ESM connected via gateway interface (nodes 2 or 9). v.) Let the OCRI and DBMS function be combined on a minicomputer with disk cartridge. The recommended approach is to supply a TI Silent 743 hard copy terminal with the FDM interfaced directly to the OCRI processor. The minicomputer approach is also very attractive; however it would represent an additional cost to the government.

7. BBSA, WBSA: This node accepts inputs from the loop to perform the BBSA and WBSA functions.

8. FIAC: This node accepts inputs from the loop to perform the FIAC function.

9. SDCA, SSCI: This node connects to the DL11E interface of the ESM PDP11/40 in loop B. In addition to the SDCA and SSCI functions it could also perform the DBMS function if the FDM required a disk cartridge for nodal station simulations. SDCA inputs can be simulated by the PDP 11/40 processor or by the B776 processor.

The column functions can be mapped to the hardware modules depending on the type of system to be simulated. A loader utility will be provided for loading software on the various microcomputers. For example for unmanned acquisition station simulation, the OCRI is omitted. Its function could be mapped onto one of the BDS microcomputers in the loop with any convenient CRT in the ESM assigned as a user terminal in order to model a remote terminal capability. Sector and nodal station simulations would use only the FIAC, DBMS, and OCRI functions.

The loop architecture shown in Figure 6-1 will perform in accordance with the requirements of the system. At a one megabit/second line rate, the speed of the loop is sufficient to handle peak loads with a safety factor of at least 2.5. This means that peak loads will be handled with average delays of not more than a few milliseconds. The absolutely maximum delay will be about 22 or 24 milliseconds. This delay will occur with very small probability. In the usual case, the delay in the srart of message delivery will be that of the write token orbit time of about 100 microseconds plus the response of the LIU giving an overall delay in the order of 110 or 120 microseconds.

Regarding redundancy, the loop itself is redundant in that it is composed of two counter-rotating loops with an automatic loop-back feature incorporated into each LIU. A detailed description of the loop-back capability is given in Appendix B. If an LIU-CP combination has a MTBF of three years and nine such nodes have a MTBF of four months, then, the single failure availability is 0.9986 based on a four hour MTTR. With loop-back, two simultaneous faults are required. This means that the system availability for LIU/CP equipments is 0.999999 with a system MTBF for the LIU/CP equipments of 4 million hours.

In the feasibility development model, the power supply and the centralized clock are not redundant. In a real system these components could be made redundant in the interests of reliability.

Since each microcomputer module is the same as any other such module (except for the program development unit) and each LIU/CP is the same as any other LIU/CP, the modules are interchangeable. The function of any such module can be taken over by a spare module. It is not planned that spare modules will be supplied with the feasibility development system, but in a real system, any number of spare modules could be supplied. Spare modules could also be supplied in the feasibility development system at extra cost.

Owing to the modularity of the system, spare modules can be added at any time. The fact that such modules exist must be added to the system records on the minidisk of the program development system, and each must be given its own function address, but logical identifiers may be mapped onto the extra modules as required.

6.3 Modules Mapped to a Bus Architecture

A typical mapping of column functions to hardware modules for an FDM connected in a serial bus configuration (e.g., ETHERNET) is shown in Figure 6-2. All the remarks of Section 6.2 concerning module functions, module interactions, and hardware requirements apply to the bus architecture. A difference is that while the loop uses an off-the-shelf Loop Interface Unit (LIU) the bus uses a Bus Interface Unit (BIU) which would take approximately 3 months and 6 man-months to develop. Also the loop can use a central clock while the bus uses individual clocks.

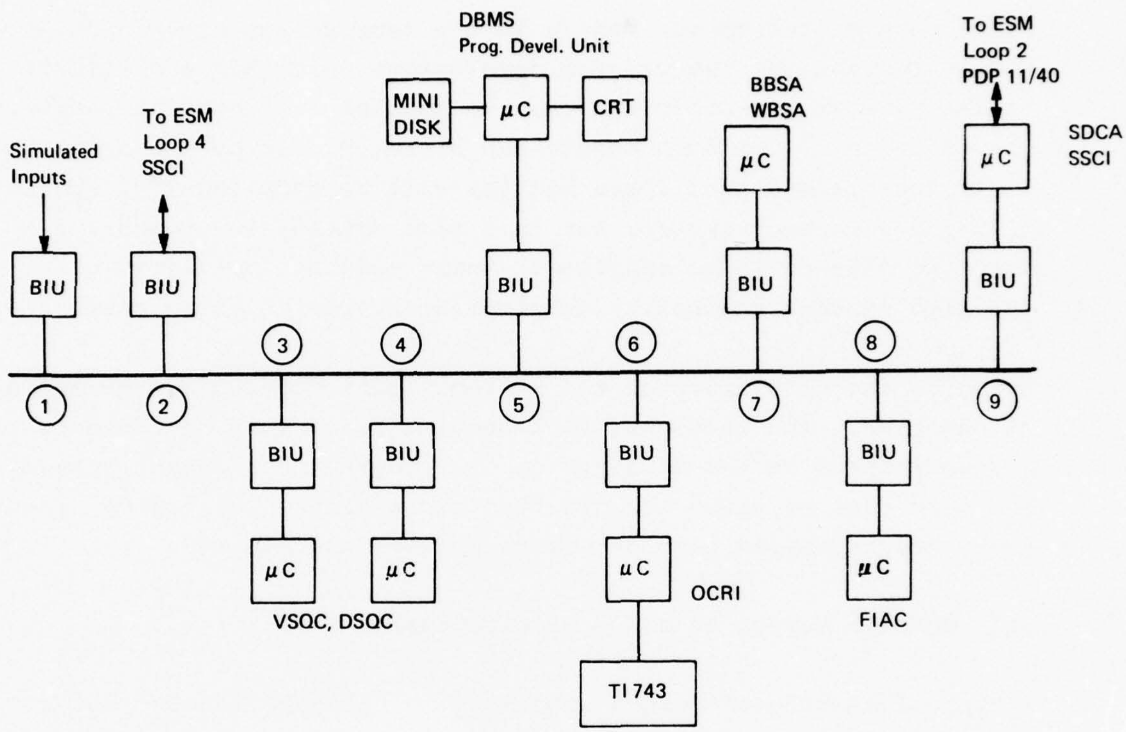


Figure 6-2
FDM Using Bus Architecture

The serial bus architecture will also perform in accordance with the requirements of the system. At the low message traffic loads of the system, the number of collisions between messages should be small so that the average delay should not be much worse than that of the loop. Since maximum delay is not bounded, however, it is possible that an occasional message delay could be troublesome.

The serial bus could be made redundant, but it does not seem to be required in a system that is enclosed within a cabinet. Where external busses are used, then bus redundancy might be needed. Since there need be no external clock, there is no problem of clock redundancy.

One of the main ideas behind the modular system control development program has been the development of a modular approach to the lower three echelons of the DCS hierarchy, i.e., the sector, node, and station levels. These levels can be implemented by interfacing different subsets of the same modules to the interprocessor communication network (i.e., loop or bus). The only difference in the modules themselves is one of size. Thus, if LSI-11 modules were used at the station level, then PDP-11 modules might be used at the node and sector levels; if TI 990/4 modules were used at the station level, then TI 990/10 modules might be used at the node and station levels.

6.4 ESM Interfacing Approach

Interfacing of the FDM to the ESM and ESMD is necessary in order to permit the modeling and simulation configurations necessary for SYSCON experiments and studies, to be performed in the Hybrid Simulation Facility at DCEC. The FDM is itself a network of nodes, or modules, each capable of accomplishing one or more of the MSCDM functions. These functions include those required at the Station, Node and Sector Level of the Syscon Hierarchy. Hence, the FDM can be used by itself to simulate all three facilities in a co-located configuration. As such the three levels interface with each

other through the interconectivity provided within the FDM. It is desirable however, to interface this FDM configuration to the ESM/ESMD, wherein the ESM/ESMD is used to simulate the ACOC. Here the FDM would interface with Loop 4, which would act as the ACOC. ESM Loop 1 would serve as the DCAOC.

Other configurations, can also be simulated by using the FDM together with the ESM/ESMD Loops. Typical arrangements, per figures 6-3, 6-4, and 6-5 include:

a) FDM as Station, Loop 4 of ESMD as node, and Loop 1 of ESM as Sector.

b) FDM as Node, Loop 1 of ESM as Sector and Loop 2 of ESM as ACOC.

c) FDM as Sector, Loop 1 of ESM as ACOC and Loop 2 of ESM as another Sector.

Note: As shown later, access to Loops 1 and 2 must be via Loop 3. Still other arrangements can be configured. In each of these, the FDM must be interfaced with the ESM/ESMD Loops. The ESM Loops are themselves interfaced such that Loops 1, 2 and 3 are each connected to the other three via gateway nodes. EMSD Loop 4, however, is connected only to Loop 3 via gateway. Loop 4 is also equipped with a gateway intended for connection to the FDM. The FDM, in turn, was intended to interface only with Loop 4 of the ESMD.

6.4.1 Communications Interface Requirements

The ATEC System Description of 1 Dec. 1976, defines the Communications Interface requirements for ATEC. These requirements are also thought to be applicable to SYSCON and, as such, are restated

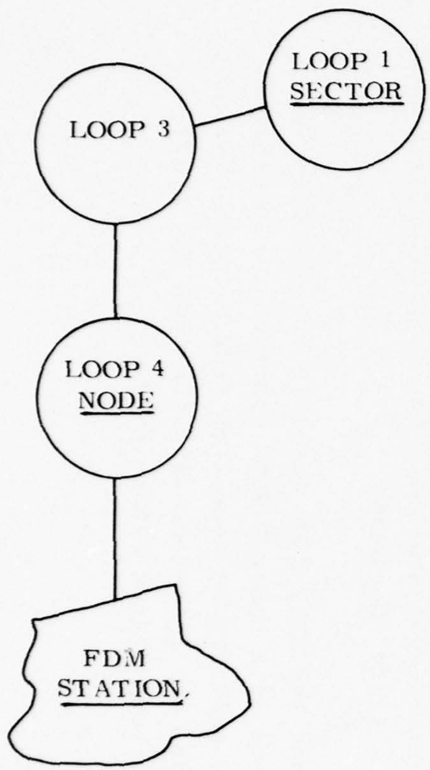


Figure 6-3
FDM as a Station

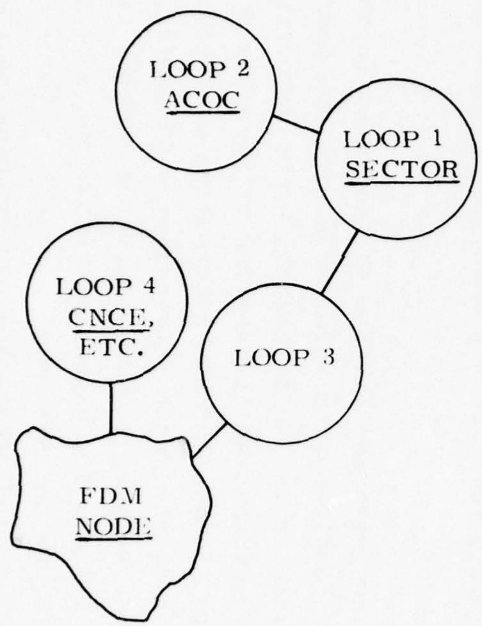


Figure 6-4
FDM as a Node

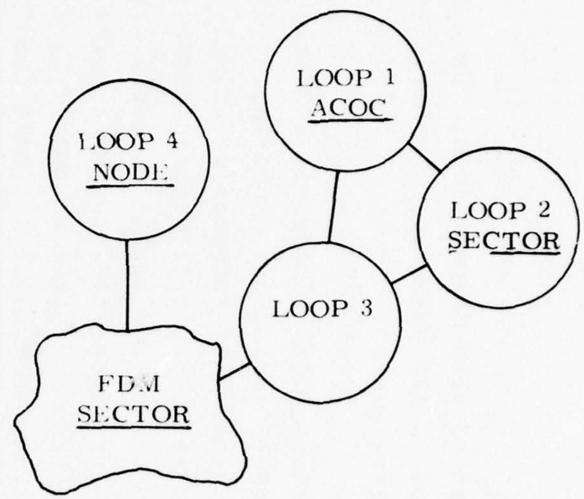


Figure 6-5
FDM as a Sector

here. They are specified with respect to Station, Node and Sector levels and are related to the specific interfaces between FDM and ESM/ESMD (see Figure 6-6).

An ATEC Station interfaces with a node via a data channel at 150 or 2400 baud, using ASCII coding and employing error detection and retransmission techniques. The information transferred between these two levels consists of control signals directing specific measurements, scan sequences and tests; measured performance parameters; status information; measurement reports; alarm notifications; data base changes; system connectivity information; and text messages between controllers. Where the FDM represents a Station and ESMD Loop 4 represents a node, their interface must satisfy the above.

An ATEC Node interfaces with the Station as indicated in the preceding paragraph. In addition, it interfaces with the Sector via a 2400 baud data channel using ASCII coding, error detection and retransmission. The information transferred between the Node and Sector Level consists of nodal status information and measured parameter information to the Sector; this same kind of information to other nodes via the Sector; requests from the Sector, or from other nodes via the Sector, for parameter measurements or tests; data base information exchange and System connectivity information between Node and Sector; and, text messages between controllers. Where the FDM represents a Node and ESM Loop 1 represents a Sector, their interface must satisfy the above.

It should be noted that the ATEC Node level is to interface with a CNCE of the TCCF, and with other SYSCON functions. Interfaces of this type are also provided by Loop 4 of ESMD. Hence, the FDM, when simulating a Node, can interface with the CNCE and other similar functions via Loop 4 (see Figure 6-4).

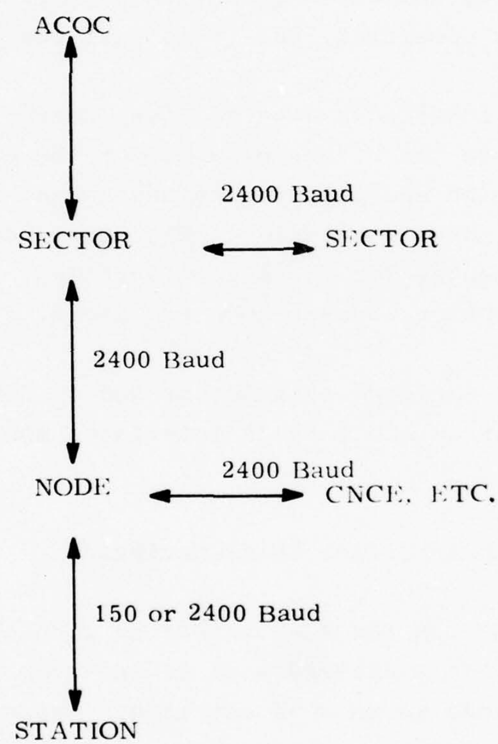


Figure 6-6
DCS Hierarchy

An ATEC Sector interfaces with the node as indicated above. In addition, it interfaces with the ACOC and with appropriate O&M Agencies. These latter interfaces provide for transmitting status reports to the ACOC and to the O&M Agency and for receipt of direction and control information from them. Transmission is to be accomplished via either a direct data channel or via AUTODIN using ASCII coding and error detection and retransmission. The data rate is not specified, but it is anticipated to be 2400 baud.

The Sector also interfaces with adjacent Sectors via 2400 baud data links. These interfaces provide for the exchange of information for fault isolation and performance assessment functions which span Sector jurisdictional boundaries. They also accomodate node-to-node information exchanges for these same purposes, as well as permitting the Sector to monitor node-to-node exchanges.

Where the FDM is employed as a Sector and an ESM Loop represents another Sector or an ACOC, their interfaces must satisfy the above requirements.

6.4.2 ESM & ESMD Interface Characteristics

The four loops of ESM and ESMD interface with each other via gateway nodes. That is a gateway node in one loop interfaces directly with a gateway node in an adjacent loop. Where the loops are co-located, the interconnect can be via twisted pair wire. Where they are widely separated, a communication medium (e.g., modems and telephone lines) can be used to interconnect them. In any case the gateways provide complete interfacing between the loops, thereby providing for protocol conversion, speed conversion, and coding conversion. Hence, the loops can be independent of each other with respect to those characteristics.

The gateway nodes are like all other nodes on the loop, except for the tailoring of the software and the minimal interface logic required for compatibility with the interconnecting communication path. Data is transmitted between gateways (i.e., between loops) in the form of a binary stream of eight bit bytes. Further it is transferred in packets of up to 256 bytes. Actual transmission rates may be up to 1 MBps.

Each of the first three loops is interfaced to each of the other two via the above described gateways (Figure 6-7). Loop 4 is interfaced to Loop 3 via the same gateway arrangement. Finally, Loop 4 is equipped with a gateway for interfacing to the FDM. It is planned that the FDM will be equipped with a compatible gateway for interfacing with the gateway of Loop 4.

6.4.3 Recommended Interfacing

In the earlier paragraphs and in Figures 6-3, 6-4, and 6-5, it was shown that the FDM requires a different interface connectivity depending upon its assignment as a Station, node or Sector.

STATION - As a Station the FDM requires connectivity only to the node which most likely would be simulated by Loop 4. In this case the FDM interfaces only with Loop 4.

NODE - As a node, the FDM requires connectivity with the Station; with the CNCE, etc., and, with the Sector. Hence, if the FDM simulates a node, it must interface with Loop 4 (which simulates the CNCE, etc.) and with Loop 1 (via Loop 3) which simulates a Sector. A Station would be simulated by the FDM. Hence, the Station-to-node connectivity would be internal to FDM.

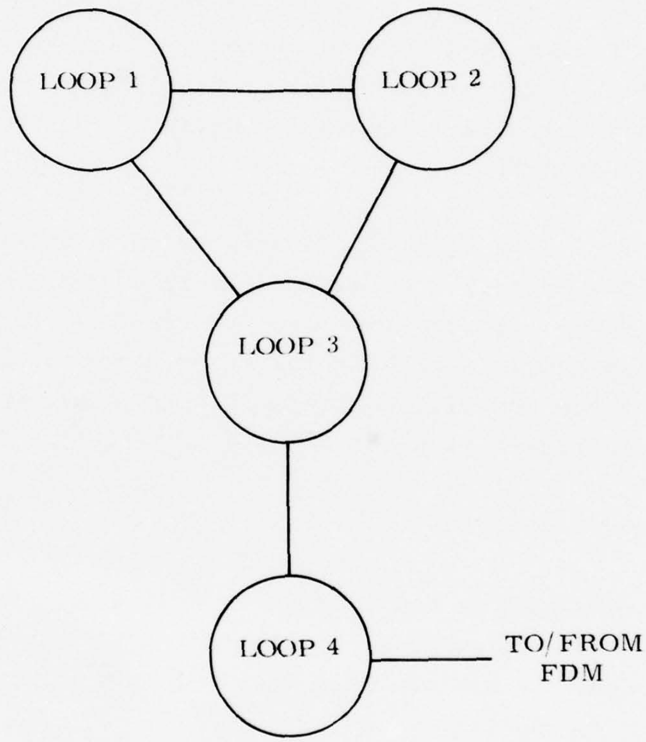


Figure 6-7
ESM Gateways

SECTOR - As a Sector, the FDM requires connectivity to the node, to the ACOC and to other Sectors. Hence, if the FDM simulates a Sector, it should interface with Loop 4 (representing a node) with Loop 1 (representing an ACOC) and with Loop 2 (representing another Sector). In actuality the interface to both Loops 1 and 2 would be accomplished via Loop 3 (See Figure 6-5).

From the above, it can be seen that for complete modeling flexibility the FDM should have a gateway to Loop 3 as well as to Loop 4, as shown in Figure 6-4 and 6-5.

In the ESMD, Loop 4 is provided with gateways to both Loop 3 and to the FDM (See Figure 6-7). However, no gateway exists in Loop 3 for connection to the FDM. Therefore the FDM-to-Loop 3 connectivity must be achieved via Loop 4. This results in considerable delays in transactions between the FDM and Loops 1 or 2, since Loop 4 as well as Loop 3 must be transited.

While it was intended that the FDM have only one gateway to the ESMD, the addition of a second gateway could provide improvement in modeling capability. This second gateway could interconnect on a time shared basis with the gateway on Loop 3 that normally connects to Loop 4. The sharing of this Loop 3 gateway could be via manually operated switch (see Figure 6-8). With the switch in one position, Loop 3 is interconnected with Loop 4. In the other position Loop 3 is interconnected with the FDM. In this latter position the interconnections illustrated by Figure 6-4 and 6-5 are provided. Now transactions between the FDM and Loops 1 and 2 must pass through Loop 3 only, rather than through both loops 4 and 3.

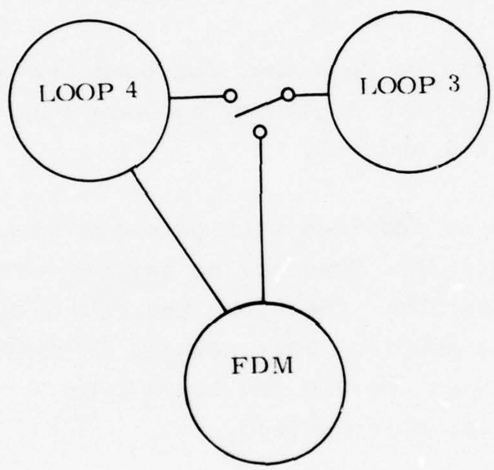


Figure 6-8
ESM - FDM Switched Connection

6.4.4 Interfacing Implementation

In addition to the above anticipated simulation configurations, it would be desirable for the FDM to have access to ESM mini-computers for simulated inputs and additional data base storage. FDM node 9 (cf. Figs. 6-1, 6-2) could have a pluggable connection to the ESM Processor B, loop 2, PDP-11/40 via a DL11-E interface. The DL11-E already exists in the PDP 11/40 host, and it is currently used for remote terminal connection via leased telephone line to Paoli. Host B might store files representing Nodal connectivity. Host B could also generate simulated switch data to the SDCA module.

Input simulation scenarios stored on the B776 disk could be transferred to the FDM via the BDS gateway of ESMD loop 4. The gateway would be connected to the LIU or BIU of node 1 (cf. Figs. 6-1, 6-2). The 2400 baud communications interface (SSCI) would connect the BDS gateway to the FDM LIU or BIU of node 2. Figure 6-9 summarizes the FDM-ESM interfacing alternatives.

6.5 Sensitivity Analysis

6.5.1 Life Cycle Costing

The section provides a comparative Life Cycle Cost (LCC) analysis for the FDM as configured with various candidate hardware architectures. The purpose of this trade-off analysis is to provide information for aiding in the selection of a cost effective approach.

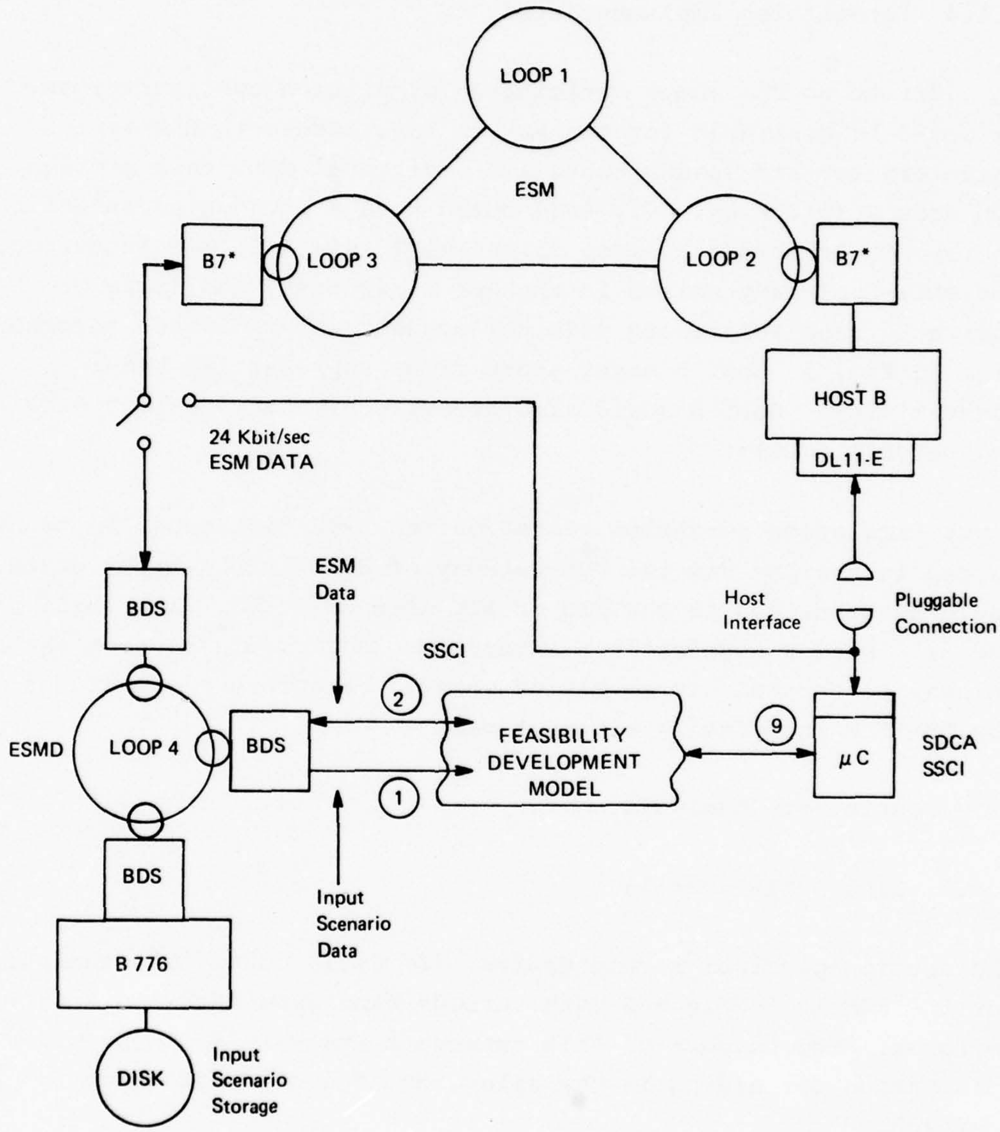


Figure 6-9
FDM-ESM Interfaces

The following paragraphs summarize the salient aspects of this analysis. All assumptions and data sources supporting the summary information contained herein have been documented and are available for reference.

6.5.1.1 Candidate Hardware Architectures

The following three hardware architectures were considered as candidates for the FDM application:

- Ring
- Serial Bus
- Parallel Bus

The following two microcomputers were considered as candidates for implementation with any of the preceding architectures:

- Texas Instrument (TI) TM990/100 with associated static memory

- Digital Equipment Corporation (DEC) KD11-F LSI-11 with associated dynamic memory

Since any of the two microcomputers can be configured with any of the three architectures, there are a total of six candidate architectural approaches to be considered.

Table 6-3 provides the building blocks for each of the six candidate architectures. It may be noted that building blocks, such as interfacing equipment and peripherals which are identical for the six architectures, have been omitted.

Table 6-3. Hardware Architecture Building Blocks

BUILDING BLOCK	HARDWARE ARCHITECTURE/BUILDING BLOCK QUANTITY					
	RING-TI	RING-DEC	SERIAL-TI	SERIAL-DEC	PARALLEL-TI	PARALLEL-DEC
1. Microcomputer						
a. TI TM 990/100	9		9		9	
b. DEC KS11-F LSI-11		10		10		10
2. Memory						
a. 64K Static	9		9		9	
b. 64K Dynamic		10		10		10
3. Arbiter Board					1	1
4. Line Interface Bus	19	20	19	20	23	25
5. Bus						
a. Connections	2	2	2	2	40	40
b. Width	1	1	1	1	40	40

The following names have been assigned to the six candidate hardware architectures for purposes of discussion:

1. Ring-TI: Ring architecture implemented with TI microcomputer TM 990/100.

2. Ring-DEC: Ring architecture implemented with DEC microcomputer KD11-F LSI-11.

3. Serial-TI: Serial Bus architecture implemented with TI microcomputer TM990/100.

4. Serial-DEC: Serial Bus architecture implemented with DEC microcomputer KD 11-F LSI-11.

5. Parallel-TI: Parallel Bus architecture implemented with TI microcomputer TM 990/100.

6. Parallel-DEC: Parallel Bus architecture implemented with DEC microcomputer KD 11-F LSI-11.

The "Ring-TI" architecture has been arbitrarily chosen as the "Baseline Architecture" for comparison with the other five candidate architectures.

6.5.1.2 LCC Analysis Methodology

In preparing this comparative analysis, computations were performed in accordance with Burroughs Corporate Standard 1257 5726 ASDET (Automated System Design and Evaluation Tools), using the mathematical models and computer programs described in the following documents:

- SLAMBDA User Instructions, Burroughs Document Number 2675 1271 dated September 15, 1975 - For MTBF predictions.

- Spares User Instructions, Burroughs Document Number 2675 1313 dated November 24, 1975 and Burroughs Spares Pipeline Model - for computation of spares cost.

- LCCA (Life Cycle Cost Analysis) User Instructions, Burroughs Document Number 2675 1354, dated April 9, 1976 - for computation of Life Cycle Costs.

The LCC profile depicted in Figure 6-10 characterizes the organization of the LCAA program output. By defining a LCC profile, time-phased projections can be made for a specified forecast period of up to sixteen years. The time-phasing and applicable parameters employed for the ADO LCC analysis are provided in subsequent paragraphs.

6.5.1.3 LCC Analysis Assumptions

For this preliminary phase of the LCC analysis, computations were limited to those cost elements associated only with non-identical building blocks (differing in quantity or types of building blocks) for the candidate FDM architectures as shown in Table 6-3. For those cost elements that are considered identical for all architectures, their values were not included, i.e., set equal to zero. In addition, for cases where cost differences occur among the candidate architectures, for certain cost elements, only the estimated differences (or Δ's) were considered and no attempt was made to estimate the overall costs.

Input
- PRODUCT NAME
- FORECAST PERIOD
- MEAN TIME BETWEEN MAINTENANCE ACTIONS(HOURS)
- AVERAGE DUTY CYCLE (%)
- MTBMA ADJUSTMENT FACTOR
- MEAN TIME TO REPAIR(HOURS)
- AVERAGE COST PER UNSCHEDULED MAINTENANCE ACTION($)
- MEAN TIME BETWEEN SCHEDULED MAINTENANCE(HOURS)
- AVERAGE COST PER SCHEDULED MAINTENANCE ATTENTION($)

a. General Product Information

1. ENGINEERING

Input
- -PRODUCT DEVELOPMENT (LABOR)
- -PRODUCT SUPPORT (LABOR)
- -TECHNIQUE STUDIES (LABOR)
- -GENERAL ENGINEERING (LABOR)
- -MATERIAL
- -MISCELLANEOUS
- -BURDEN (%)-REF.
- -GENERAL & ADMINISTRATIVE(%)-REF.

Computed
- -BURDEN
- -GENERAL & ADMINISTRATIVE
- SUBTOTAL (ANNUAL)
- SUBTOTAL (CUMULATIVE)

2. MANUFACTURING

Input
- -DIRECT LABOR
- -MATERIAL
- -WHOLE OVERHEAD
- -ASSETS
- -MISCELLANEOUS
- -QUANTITY MANUFACTURED-REF.
- -BURDEN (%)-REF.
- -GENERAL & ADMINISTRATIVE(%)-REF.

Computed
- -BURDEN
- -GENERAL & ADMINISTRATIVE
- SUBTOTAL (ANNUAL)
- SUBTOTAL (CUMULATIVE)

3. FIELD SUPPORT

Input
- -TRAINING (LABOR)
- -INSTALLATION (LABOR)
- -MODIFICATIONS (LABOR)
- -DOCUMENTATION (LABOR)
- -TRAINING (MATERIAL)
- -INSTALLATION (MATERIAL)
- -MODIFICATIONS (MATERIAL)
- -DOCUMENTATION (MATERIAL)
- -CAPITAL EQUIPMENT
- -MISCELLANEOUS
- -QUANTITY MAINTAINED-REF.
- -BURDEN (%)-REF.
- -GENERAL & ADMINISTRATIVE(%)-REF.

Computed
- -UNSCHEDULED MAINTENANCE
- -SCHEDULED MAINTENANCE
- -BURDEN
- -GENERAL & ADMINISTRATIVE
- SUBTOTAL (ANNUAL)
- SUBTOTAL (CUMULATIVE)
- TOTAL (ANNUAL)
- TOTAL (CUMULATIVE)

b. Major Cost Categories

Figure 6-10. LCC Profile

6.5.1.3.1 Maintenance Philosophy

Due to the lack of information regarding costs associated with site and depot repair by government personnel, the following assumptions have been made concerning the general maintenance philosophy of the DCS network:

- Maintenance performed by Burroughs personnel
- DCS network consists of 1000 sites
- 250 Field Engineering maintenance locations
- Repair performed on site by removing and replacing failed boards
- All failed boards (with the exception of DEC's microcomputer) returned to Burroughs central facility for repair.
- DEC microcomputer boards returned to DEC for repair. This is necessitated because of DEC's policy which prohibits the purchase of spare LSI's separately.

6.5.1.3.2 Time-Phasing

Figure 6-11 depicts the preliminary time-phasing assumed for the DCS network over a period of twelve years. This time phasing represents the major cost categories (engineering, manufacturing, and field support) and their relative occurrence in time. The build-up of equipment with time as assumed for this analysis is also shown in Figure 6-11 under the Field Support Phase.

All costs in this analysis are escalated on a yearly basis to represent time-phased costs. Pertinent assumptions associated with the specific cost parameters used in this analysis for a particular cost category are provided in Paragraph 6.5.1.6.

ENGINEERING PHASE

MANUFACTURING PHASE

	1979	1980	1981	1982	1983	1984	1985
YEARLY			200	200	200	200	200
CUMULATIVE			200	400	600	800	1000

QUANTITY MANUFACTURED

FIELD SUPPORT PHASE

	1979	1980	1981	1982	1983	1984	1985	1986	1987	1988	1989	1990
YEARLY CHANGE			200	200	200	200	200	0	0	0	0	0
CUMULATIVE			200	400	600	800	1000	1000	1000	1000	1000	1000
EQUIPMENT YEARS			200	600	1200	2000	3000	4000	5000	6000	7000	8000

QUANTITY MAINTAINED

OPERATIONAL YEAR			1	2	3	4	5	6	7	8	9	10
	1979	1980	1981	1982	1983	1984	1985	1986	1987	1988	1989	1990

CALENDER YEAR

Figure 6-11. 12-Year Time-Phasing

Table 6-4. LCC Comparison Summary

COMPARISON OF LIFE CYCLE COSTS ----- 1000 UNITS .... DATE: 08/25/77
(IN THOUSANDS OF DOLLARS)

FORECAST PERIOD ...... 1979 -- 1990 (12 YEARS)

CASE	RUN IDENTIFICATION (CASE 1 IS BASELINE)	COST CATEGORY: ENGINEERING COST	ENGINEERING % OF TOTAL LCC	ENGINEERING DELTA BASELINE	MANUFACTURING COST	MANUFACTURING % OF TOTAL LCC	MANUFACTURING DELTA BASELINE	FIELD SUPPORT COST	FIELD SUPPORT % OF TOTAL LCC	FIELD SUPPORT DELTA BASELINE	TOTAL LIFE CYCLE COST: COST	TOTAL LIFE CYCLE COST: DELTA BASELINE
1	RING-TI	0	0.0	-	68073	47.6	-	74798	52.4	-	142871	-
2	RING-DEC	0	0.0	0	85065	48.6	16992	90090	51.4	15292	175155	32284
3	SERIAL-TI	90	0.1	90	68073	47.6	0	74798	52.3	0	142961	90
4	SERIAL-DEC	90	0.1	90	85065	48.5	16992	90090	51.4	15292	175245	32374
5	PARALLEL-TI	180	0.1	180	75311	49.4	7238	77041	50.5	2243	152532	9661
6	PARALLEL-DEC	180	0.1	180	93607	50.2	25534	92662	49.7	17864	186450	43579

6.5.1.4 Comparison of Results

Table 6-4 summarizes the comparative life cycle costs as projected for the six candidate architectures comprising the building blocks contained in Table 6-3. As indicated previously, the Ring-TI architecture has been arbitrarily selected as the baseline and is reflected as Case 1 in Table 6-4. The cost of each major category, as well as the total LCC are shown for each of the candidate architectures. In addition, the cost differences, relative to the baseline, are provided. The unsigned DELTA values indicate that the cost is in excess of the baseline cost by the amount specified. These DELTAS, in effect, represent the overall cost differences associated with the totality of equipment in the DCS network when implemented with a particular architecture. The costs for the equipments that are common to all architectures are not included since they would not impact the results of this comparative analysis.

6.5.1.5 LCC Ranking

Table 6-5 provides the life cycle cost ranking based on the forecasted LCC costs in Table 6-4, when the DCS network is implemented with the six candidate architectures.

Table 6-5. LCC Ranking

RANKING	CANDIDATE HARDWARE ARCHITECTURE
Best	Ring-TI
Second	Serial-TI
Third	Parallel-TI
Fourth	Ring-DEC
Fifth	Serial-DEC
Worst	Parallel-DEC

6.5.1.6 Input Data Summary

This paragraph contains a summary of the sources, assumptions, and related input data for the six candidate architectures. Table 6-6 delineates the basis for estimating each of the LCAA input parameters. These input parameters are provided in the order shown for the LCC Profile in Figure 6-3. The numbers (1 through 6) referenced in Table 6-6 correspond to the architecture types as defined in paragraph 6.5.1.1, i.e., 1 = Ring TI, 2 = Ring-DEC, etc. The "X's" below the architecture type in Table 6-6 indicate the "INPUT/REMARKS" that correspond to a particular architecture.

As indicated previously, input parameters for this analysis have been limited to Engineering, Manufacturing, and Field Support costs associated only with the non-identical building blocks (differing in quantity or types of building blocks) in Table 6-3. Even for these non-identical building blocks, wherever the costs, (such as documentation labor costs, etc.) were identical, their value has been assumed zero. In addition, in the case of costs such as "Engineering Development Labor", only the cost differences among the architectures have been considered.

It may be noted that for those cases where the values were set equal to zero, descriptions of the input parameters are provided for reference.

Table 6-6. Input Data Summary (Sheet 1 of 6)

COST CATEGORY	REQUIRED LCCA INPUT	ARCHITECTURE TYPE 1	2	3	4	5	6	INPUT/REMARKS	
GENERAL	Product Name	X						Ring-TI	Reference: Paragraph 6.5.1.1
			X					Ring-DEC	
				X				Serial-TI	
					X			Serial-DEC	
						X		Parallel-TI	
							X	Parallel-DEC	
	Forecast Period	X	X	X	X	X	X	1979 through 1990, as shown in Figure [illegible].	
	Mean Time Between Maintenance Actions (MTBMA or MTBF)	X						666 Hours	MTBF Prediction at 35°C ambient and Ground Benign environment.
			X					572 Hours	
				X				666 Hours	
					X			578 Hours	
						X		625 Hours	
							X	542 Hours	
	Average Duty Cycle	X	X	X	X	X	X	Projected at 100% - Continuous operation.	
	MTBMA Adjustment Factor	X	X	X	X	X	X	Not Used - Set equal to one throughout,	
	Mean Time To Repair	X	X	X	X	X	X	Estimated as 0.75 hours (Actual repair time; does not include logistic/administrative delays, if any).	
	Average cost per Unscheduled Maintenance Action (less spares and spares transportation costs)	X		X		X		$180.24	Note: This input parameter includes the cost at site to replace the faulty board and the cost of repair of faulty board at a central facility. The two costs are further broken down as shown below:
			X		X			$190.[illegible]	
							X	$189.73	

Site Repair:

(a) Cost of field engineer's time

(i) An average two way travel time of two hours

(ii) An average repair time of 0.75 hours

(b) Travel (milage) cost for an average two way travel of 40 miles

Board Repair [illegible] Central Location (Depot)

(a) Cost of repair of each board

(b) Cost of two way surface transportaion for the faulty board to and from the central repair facility.

Costs for the repair of the boards in Architectures 1, 3, and 5 were based on 1977 budgeting costs which were then escalated (at 6% per year from 1977) to arrive at average escalated costs applicable for the period 1981 through 199[illegible]. Repair costs for the boards in Architectures 2, 4, and 6 were computed

Table 6-6. Input Data Summary (Sheet 2 of 6)

COST CATEGORY	REQUIRED LCCA INPUT	ARCHITECTURE TYPE 1	2	3	4	5	6	INPUT/REMARKS
GENERAL	Average cost per Unscheduled Maintenance (Cont.)							in a similar way. However, in this case, Digital Equipment Corporation's (DEC) estimate for (depot) repair of their microcomputer board was taken into account in addition to the Burroughs cost estimates for site repairs for all other equipment.
	Mean Time Between Scheduled Maintenance	X	X	X	X	X	X	Not Applicable; equipment in Table 6.5.1 does not require any scheduled/ Preventive Maintenance.
	Average cost per Scheduled Maintenance							Zero; no scheduled / Preventive Maintenance is required.
ENGINEERING	Associated Years	X	X	X	X	X	X	1979 through 1985, as shown in Figure
	Product Development Labor	X	X					Zero } Δ's shown, Actual costs not considered
				X	X			$40,000.00 in 1979 } Δ's shown, Actual costs not considered
						X	X	$80,000 in 1979 Note: Product Development effort spans 1979 and 1980. The serial and parallel Bus architectures require $40,000 and $8[illegible] more than that for Ring architecture during the first year (1979).
	Product Support Labor	X	X	X	X	X	X	Assumed Zero; as identical leveled-off Engineering costs are required to support Manufacturing and Field Engineering during years 1981 through 1985 for the six architectures.
	Technique Studies Labor	X	X	X	X	X	X	Zero; Not Applicable.
	General Engineer-Labor	X	X	X	X	X	X	Zero; Not Applicable.
	Material	X	X	X	X	X	X	Assumed Zero; as these costs are identical for the six candidate architectures. This cost parameter includes costs for brass boards, travel costs, EMI material costs, computer time costs, technical manual costs, and test documentation costs.
	Miscellaneous	X	X	X	X	X	X	Assumed Zero; as these costs are identical for the six candidate architectures. This cost parameter includes the costs for test equipment.
	Burden	X	X	X	X	X	X	92.4%.
	General and Administrative	X	X	X	X	X	X	17%.

Table 6-6. Input Data Summary (Sheet 3 of 6)

COST CATEGORY	REQUIRED LCCA INPUT	ARCHITECTURE TYPE 1	2	3	4	5	6	INPUT/REMARKS
MANUFACTURING	Associated Years	X	X	X	X	X	X	1981 through 1985, as shown in Figure [illegible].
	Direct Labor	X						$479,741, $508,526, $539,037, $571,379, and $605,662 for years 1981 through 1985, respectively.
			X					$504,991, $535,290, $567,408, $601,452, and $637,539 for years 1981 through 1985, respectively.
				X				$479,741, $508,526, $539,037, $571,379, and $605,662 for years 1981 through 1985, respectively.
					X			$504,991, $535,290, $567,408, $601,452, and $637,539 for years 1981 through 1985, respectively.
						X		$617,351, $654,392, $693,656, $735,275, and $ 779,392 for years 1981 through 1985, respectively.
							X	$669,113, $709,260, $751,815, $796,924, and $844,739 for years 1981 through 1985, respectively.
								Note: The above costs are based on 1977 budgetary estimates (escalated at 6% per year to the applicable year). These costs do not cover the vendor-furnished boards. The total costs of the vendor furnished boards have been included in the next input parameter "MATERIAL".
	Material	X						$8,388,932, $8,892,268, $9,425,80[illegible], $9,991,352, and $10,590,833 for years 1981 through 1985, respectively.
			X					$10,863,562, $11,515,375, $12,206,297, $12,938,674, and $13,714,994 for years 1981 through 1985, respectively.
				X				$8,388,932, $8,892,268, $9,425,804, $9,991,352, and $10,590,833 for years 1981 through 1985, respectively.
					X			$10,863,562, $11,515,375, $12,206,297, $12,938,674, and $13,714,994 for years 1981 through 1985, respectively.
						X		$8,932,110, $9,468,037, $10,036,118, $10,638,285, and $11,276,582 for years 1981 through 1985, respectively.
							X	$11,497,724, $12,187,592, $12,918,847, $13,693,977, and $14,515,615 for years 1981 through 1985, respectively.
								Note: The above costs are based on 1977 budgetary estimates (escalated at 6% per year to the applicable year) for Burroughs boards in Figure 6.5.1-1. Costs for vendor-furnished boards were obtained from the respective vendors and were also escalated at 6% per year from 1977. The total costs were also adjusted for a 4.2% Burden on the materials.

Table 6-6. Input Data Summary (Sheet 4 of 6)

COST CATEGORY	REQUIRED LCCA INPUT	ARCHITECTURE TYPE 1	2	3	4	5	6	INPUT/REMARKS
MANUFACTURING (Cont.)	Assets	X	X	X	X	X	X	Assumed Zero: as these costs are identical for the six candidate architectures. This cost parameter includes the costs for special test equipment.
	Miscellaneous	X	X	X	X	X	X	Assumed Zero; as these costs are identical for the six candidate architectures. This cost parameter includes the Manufacturing start-up costs.
	Quantity Manufactured	X	X	X	X	X	X	200 sets per year from 1981 through 1985 assumed.
	Burden	X	X	X	X	X	X	302.8 % for Manufacturing Labor.
	General and Administrative	X	X	X	X	X	X	17 %.
FIELD SUPPORT	Associated Years	X	X	X	X	X	X	1981 through 1990, as shown in Figure [illegible].
	Training Labor	X	X	X	X	X	X	Assumed Zero: as these costs are identical for the six candidate architectures. This cost parameter includes the labor costs for training of Operators and Field Engineers.
	Installation Labor	X	X	X	X	X	X	Assumed Zero: as these costs are identical for the six candidate architectures. This cost parameter includes the labor costs for installation of the equipment in Table 6.5.1 at government sites.
	Modification Labor	X	X	X	X	X	X	Assumed Zero: as these costs are identical for the six candidate architectures. This cost parameter includes the labor cost of modification, if any, at sites during initial years.
	Documentation Labor	X	X	X	X	X	X	Assumed Zero: as these costs are identical for the six candidate architectures. This cost parameter includes the labor costs for compilation of following documentation: • Equipment Installation Guides • Field Engineering Technical Manuals • Parts Catalogs • Reference/System Operating Guides • Logic Improvement Notice/Reliability Improvment Notice/Advanced Technical Information • Performance Oriented Training Manuals.

Table 6-6: Input Data Summary (Sheet 5 of 6)

COST CATEGORY	REQUIRED LCCA INPUT	ARCHITECTURE TYPE 1	2	3	4	5	6	INPUT/REMARKS
FIELD SUPPORT (Cont.)	Training Material	X	X	X	X	X	X	Assumed Zero; as these costs are identical for the six candidate architectures. This cost parameter includes the material/printing/reproduction costs of Performance Oriented Training Manuals for operators and field engineers.
	Installation Material	X	X	X	X	X	X	Assumed Zero: as these costs are identical for the six candidate architectures. This cost parameter includes the costs of materials for Installation of equipment in Table 6.5.1 at government sites.
	Modification Material	X	X	X	X	X	X	Assumed Zero: as these costs are identical for the six candidate architectures. This cost parameter includes the costs of integrated circuits/other components for carrying out the modifications, if any, during the intital year at the government sites.
	Documentation Material	X	X	X	X	X	X	Assumed Zero; as these costs are identical for the six candidate architectures. This cost parameter includes the material/printing/reproduction costs for following documentation: • Equipment Installation Guides • Field Engineering Technical Manuals • Parts Catalogs • Reference/System Operating Guides • Logic Improvement Notice/Reliability Improvement Notice/Advance Technical Information • Performance Oriented Training Manuals.
	Capital Equipment	X	X	X	X	X	X	Assumed Zero; as these costs are identical for the six candidate architectures. This cost parameter includes the costs of using the Burroughs Digital Meter (BDM) by the Field Engineers for fault isolation to a single board.
	Miscellaneous (Spares and Spares Transportation Costs)	X						These costs are for years 1981 through 1990, respectively. $12,393,986, $9,633,411, $10,244,474, $10,894,185, $11,584,981, $197,149, $208,977, $221,516, $234,807, and $248,896.
			X					$14,591,045, $11,651,738, $12,385,129, $13,164,581, $13,992,981, $204,512, $216,783, $229,790, $243,578, and $258,192.
				X				$12,393,986, $9,633,411, $10,244,474, $10,894,185, $11,584,981, $197,149, $208,977, $221,516, $234,807, and $248,896.

Table 6-6. Input Data Summary (Sheet 6 of 6)

COST CATEGORY	REQUIRED LCCA INPUT	ARCHITECTURE TYPE 1	2	3	4	5	6	INPUT/REMARKS
FIELD SUPPORT (Cont.)	Miscellaneous (Spares and Spares Transportation Costs) (Continued)				X			\$14,591,045, \$11,651,738, \$12,385,129, \$13,164,581, \$13,992,981, \$204,512, \$216,783, \$229,790, \$243,578, and \$258,192.
						X		\$12,780,192, \$9,770,975, \$10,390,659, \$11,049,531, \$11,750,061, \$199,354, \$211,315, \$223,994, \$237,434 and \$251,680.
							X	\$15,005,463, \$11,819,257, \$12,563,129, \$13,353,716, \$14,193,948, \$207,090, \$219,516, \$232,687, \$246,648, and \$261,447.
								Note: This cost parameter includes the Repairable Spares (boards) costs at 250 Burroughs field engineering locations and the throwaway (components) spares costs at a central location (Board Repair Facility). Component Spares for Digital Equipment Corporation's microcomputer board KD11-F LSI-11 are not included herein. As mentioned earlier this board will be sent to Digital Equipment Corporation for repair and return. All spares computations are based on a 90% confidence level.
	Quantity Maintained	X	X	X	X	X	X	200, 400, 600, 800, 1000, 1000, 1000, 1000, 1000, and 1000 ~~774, 735, and 698~~ for years 1981 through 1990, respectively, as shown in Figure A-2. Note: This input parameter, along with "MTBMA" and "Average Cost per Unscheduled Maintenance Action", is used by the Burroughs LCCA program to compute yearly and cumulative costs of repairs at sites (by board replacement) and the board repairs at a central repair facility.
	Burden	X	X	X	X	X	X	Zero; Not Applicable.
	General and Administrative	X	X	X	X	X	X	Zero; Not Applicable.

6.5.2 Loop vs. Bus Architecture

The material that follows is a comparison between loop and bus architectures. The differences are stressed. Similarities such as the fact that both are flexible, adaptable and reliable are understood.

To lay the groundwork for throughput comparison, a queuing theory analysis is supplied in 6.5.2.1 for both the WT-2 loop (as the worst of the loops for throughput) and the bus. Simulation models and results are given in 6.5.2.2.

A discussion of the various differences is given in 6.3.2.3 through 6.5.2.6. The discussion is not limited to small in-cabinet systems such as the Feasibility Development Model.

6.5.2.1 Queuing Analysis

Both the WT-2 loop and the bus architectures show a considerable capability to support high throughputs with very low queuing provided that the number of nodes is reasonably large and that the traffic load is evenly distributed among the nodes. The loop architecture shows a somewhat smaller queue size than that of the bus for traffic loads in the region of 80 to 90 percent of maximum load. An analysis of the queuing situation is given below. This is followed by a comparison of simulation runs at a traffic load of about 85 per cent of maximum.

The analysis assumes a constant packet size and a constant service time T. If the Poisson arrival rate of packets to the system is λ, then the load factor ρ is given by the product λT. If there are N nodes, then the Poisson arrival into each node is λ/N. Each node is a single server with only one packet allowed at its inter-

face to the loop or bus. Any others that are in the node are in queue. The average queue length, L_N, is to be determined.

There can be at most N packets at the interfaces at any time since there are only N interfaces. This is a limited queue situation that can be described by the equation

$$L = \rho \frac{1-X^N}{1-X} \qquad (6.5.2\text{-}1)$$

$$X = \frac{\rho}{2-\rho} \qquad (6.5.2\text{-}2)$$

where X is a modified load factor that takes the constancy of T (not exponentially distributed T) into account and L is the average number of packets at the nodal interfaces. Note that as ρ approaches unity, X approaches unity and L approaches N. Table 6-7 shows a plot of L as a function of ρ for N=10.

The effective utilization factor of each node interface is given by

$$\rho_N = \frac{L}{N} \qquad (6.5.2\text{-}3)$$

Note that thus far, the analysis applies equally to the loop and the bus.

The main difference between the loop and the bus is in the standard deviation that applies to the total service time for the nodal interface. This nodal interface service time includes the wait to be served as well as the packet service time T itself. In the case of the bus, the order of service is random so that the ratio of standard deviation to total service time is unity. In the case

Table 6-7
System with 10 Nodes
Load Factor versus Node Interfaces Occupied

Load Factor ρ	Node Interfaces Occupied L	Interface Load Factor ρ_N
1.00	10.000	1.000
0.98	8.240	0.824
0.96	6.875	0.688
0.94	5.806	0.581
0.92	4.961	0.496
0.90	4.285	0.429
0.88	3.738	0.374
0.86	3.292	0.329
0.84	2.924	0.292
0.82	2.617	0.262
0.80	2.358	0.236
0.75	1.863	0.186
0.70	1.514	0.151

of the loop, the order of service is in fixed sequence so that the ratio of standard deviation to total service time is less than unity. The ratio for the loop approaches zero as ρ approaches unity. The ratio at a ρ of 0.85 is 0.67. The queue size L_{NB} for the bus is given by

$$L_{NB} = \frac{\rho_N^{\,2}}{1-\rho_N} \tag{6.5.2-4}$$

The queue size L_{NL} for the loop is given by

$$L_{NL} = \frac{\rho_N^{\,2}}{1-\rho_N} \quad \frac{1+R^2}{2} \tag{6.5.2-5}$$

where R is the standard deviation ratio. If R is considered linear with ρ with a value of 0.0 at ρ=1 and a value 0.67 at ρ = 0.85 then a comparison is given for the bus and loop queue lengths in table 6-8.

For a specific example, assume a packet of 250 characters requires 200 time units for service plus another 8 time units for ACK-NAK transmittal plus 6 time units for write token time for the loop or 3 time units for bus service overhead. This gives a total of 214 time units for the loop and 211 time units for the bus. The load factors and node interface queue lengths are given in table 6-9 for various average interarrival intervals.

6.5.2.2 Simulation Models and Results

Both the Loop and the Serial Bus were simulated for a system of 10 nodes with approximately 85% utilization using the BOSS (Burroughs Operational Systems Simulator) and the results compared. The final reports for each architecture are given in Appendix C. To perform the simulation, the two architectures were modelled in BOSS parlance. A short description of each model followed by the results are given.

AD-A063 369 BURROUGHS CORP PAOLI PA FEDERAL AND SPECIAL SYSTEMS GROUP F/G 17/2
MODULAR SYSTEM CONTROL DEVELOPMENT MODEL (MSCDM). SECTIONS 1-7.(U)
SEP 77 DCA100-76-C-0083
UNCLASSIFIED 64280 SBIE-AD-E100 136 NL

3 OF 3
AD
A063369

END
DATE
FILMED
3--79
DDC

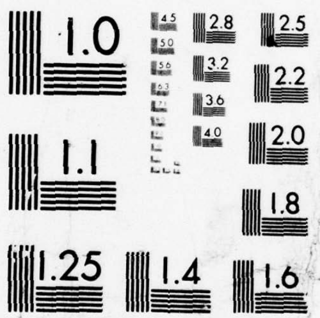

1.0
4.5
5.0
5.6
6.3
2.8
3.2
3.6
4.0
2.5
2.2
2.0
1.1
1.8
1.25
1.4
1.6

Table 6-8
System with 10 Nodes
Bus and Loop Interface
Queue Lengths

Load Factor	Queue Lengths	
	Bus	Loop
ρ	L_{NB}	L_{NL}
1.00	∞	∞
.98	3.857	1.940
.96	1.517	0.782
.94	0.806	0.431
.92	0.488	0.274
.90	0.322	0.193
.88	0.223	0.143
.86	0.161	0.111
.84	0.120	0.091
.82	0.093	0.076
.80	0.073	0.066
.75	0.043	0.042
.70	0.037	0.027

Table 6-9
Interface Queue Length
Comparison Loop vs Bus

Packet Interarrival Interval	Load Factors		Interface Queue Lengths	
	Bus	Loop	Bus	Loop
Time Units	ρ_B	ρ_L	L_{NB}	L_{NL}
235	0.898	0.911	0.310	0.233
240	0.879	0.892	0.220	0.170
245	0.861	0.873	0.165	0.130
250	0.844	0.856	0.130	0.105
255	0.827	0.839	0.103	0.089
260	0.811	0.823	0.083	0.077

6.5.2.2.1 Loop Model

BOSS models are described in terms of "processes" each of which is composed of one or more "tasks" that are grouped in a logical sequence. Each task uses time, resources or both. Time is represented in terms of "time units" where a time unit is the lowest unit of simulated time duration. It may represent a nanosecond or a millenium at the whim of the modeller but is constant for a given simulation. Resources are represented by "units". Each unit has its own identifying number. "Unit types" are pools of units wherein each unit within a type is considered to have the same characteristics as any other unit within the type. Thus, for example, unit type 7 may be considered to be a pool of tires. The units that are members of the pool are numbered 14, 21, 3, 6, 231 and 63. Each unit is a tire and when a task calls for a unit of type 7, one of the units (say that numbered 21) will be "busy" for the duration (in time units) of the task. When all six are busy simultaneously, a further call on type 7 will cause the calling task to queue on type 7.

Sometimes it is convenient to generate tasks that use neither time nor resources. These tasks are called "dummies". They are used to provide special functions that tasks can generate independently of the use of time or resources.

For the WT-2 loop model, processses 1 through 10 are used to represent the action of a packet through a node from its arrival at the node to its delivery from the node to the system. The process number represents the node number. Processes 1-10 are diagrammed in Figure 6-12. Each process starts with task 1. Task 1 uses no time but "seizes" the single unit in unit type N+10. A seized unit remains in use until the end of another task (in this case task 2). Any other start of process N will have its task 1 queue on unit type N+10 until the unit is released. The queue is first in-first out. Thus, the unit type N+10 may be considered to represent the loop interface of node N.

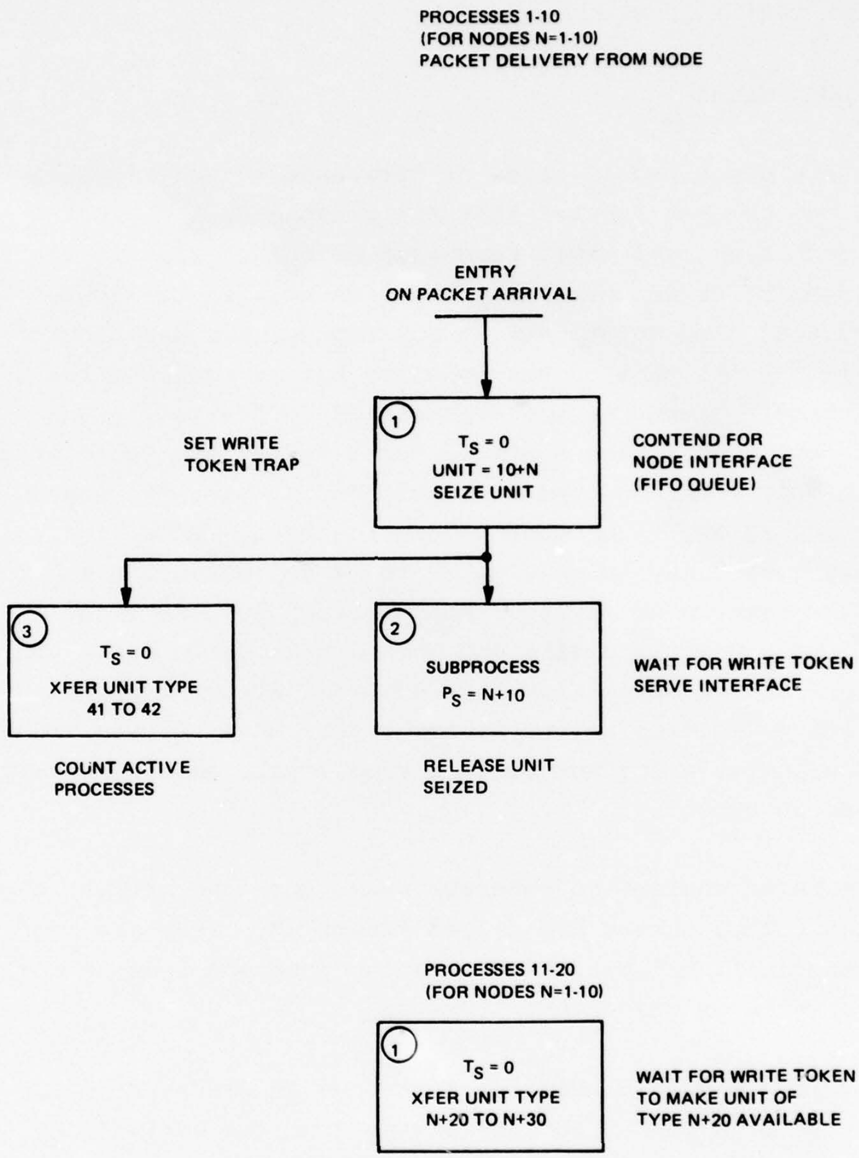


Figure 6-12

BOSS Processes 1-20 Loop Simulation

The completion of task 1 (as soon as the unit is seized) starts both tasks 2 and 3. Task 3 transfers a unit from unit type 41 to unit type 42. Initially unit type 41 holds 10 units and unit type 42 holds none. Task 3, by transferring a unit, keeps a count of processes 1-10 that have seized their respective node interface units. This is done only to save computer time as will be seen later. Task 2 calls a "subprocess" number N+10. A subprocess is a process that is called by a task. Any process can be called this way and the calling task lasts as long as the subprocess. The calling of subprocesses has many uses. In this case, the use is the provision of separate duration statistics for write token service.

Processes 11-20 are the subprocesses called by tasks 2 of processes 1-10. Each is a single task of zero duration that waits for a unit of type N+20 which is not available until the orbiting write token (process 21) makes it so. As soon as the unit is available, the task takes it and transfers it to type N+30 thereby making it not available for the next occurrence of process N+10.

Thus, processes 1-10 and 11-20 take no time of their own. All of their time is in queue either waiting for a unit of type N+10 (the node interface) to be available or of type N+20 (activated by the orbiting write token) to be available.

The key process then is process 21 the write token orbiter. The process is shown in part in Figure 6-13. There are 10 sets of 3 tasks; N, N+10, N+20 plus a common task 31. At the entry to each set is a decision point depending on whether the unit of type N+10 (the node interface) is available. If yes, then the interface is not busy and task N is started. This uses the unit for 1 time unit (the free write token time). If the unit is not available, then the interface is busy and task N+10 is started.

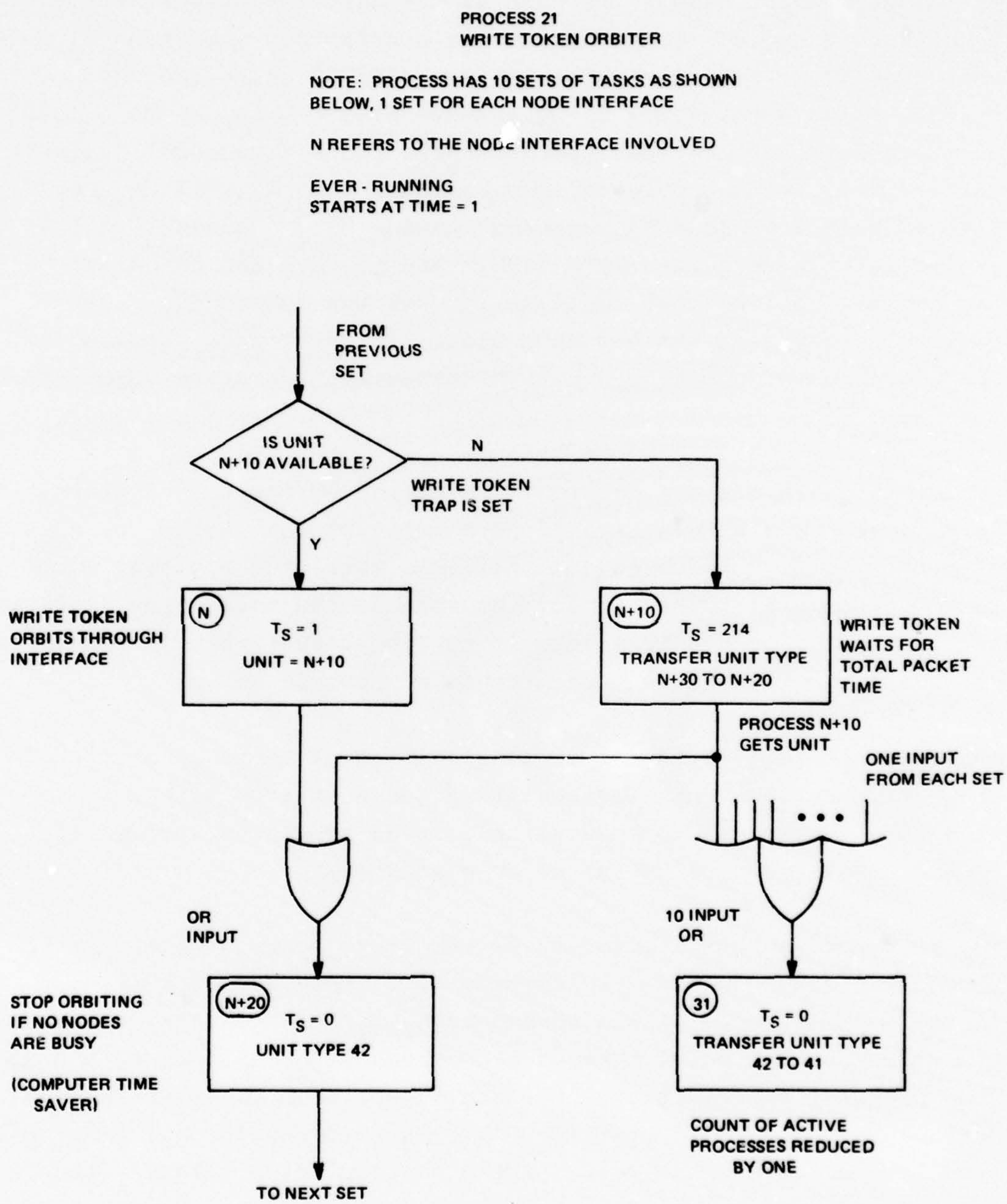


Figure 6-13
BOSS Process 21 Loop Simulation

Task N+10 makes a unit of type N+30 busy for the total packet and ACK/NAK + write token IN/OUT time of 214 time units. The unit is then transferred to type N+20 whereupon it becomes available to process N+10.

When either of task N or N+10 is ended, then task N+20 is started. This task uses a unit of type 42 for no time. If a unit of type 42 is not available (there are no active node interfaces) then the orbiting stops. This saves considerable computer time. Task 31 is started at every completion of a task N+10. Task 31 removes a unit of type 42 and transfers it to type 41. This keeps the count of active node interfaces accurate.

Process 22 (Figure 6-14) is the distributor of packet arrivals to the various nodes. A process of this type is started at a Poisson interval of 250 time units. The various tasks set up an equally weighted Monte Carlo choice of one from the processes 1-10 as a sub-process of process 22. Thus, there is a process 22 for each process 1-10 and the duration statistics for process 22 is the conglomerate of those for processes 1-10.

6.5.2.2.2 Results of Loop Simulation

Processes 1-10 show numbers of completions varying from 950 to 1061 giving a total of 10125 and a standard deviation of 32 which is quite close to the expected value of 31.* The total of 10125 has a deviation of 125 from the expected value which is close to the expected standard deviation of 100. Thus, the actual values of completions is about as expected. Process 22, the conglomerate of all processes 1-10 shows a completion time histogram of 214 time units minimum (as expected), a median value of 589 time units, a mean value of 915 time units and a standard deviation of 929 time units. The ratio of standard deviation to mean was 1.015.

---

*31 is approximately the square root of 1000 (the expected completions per node).

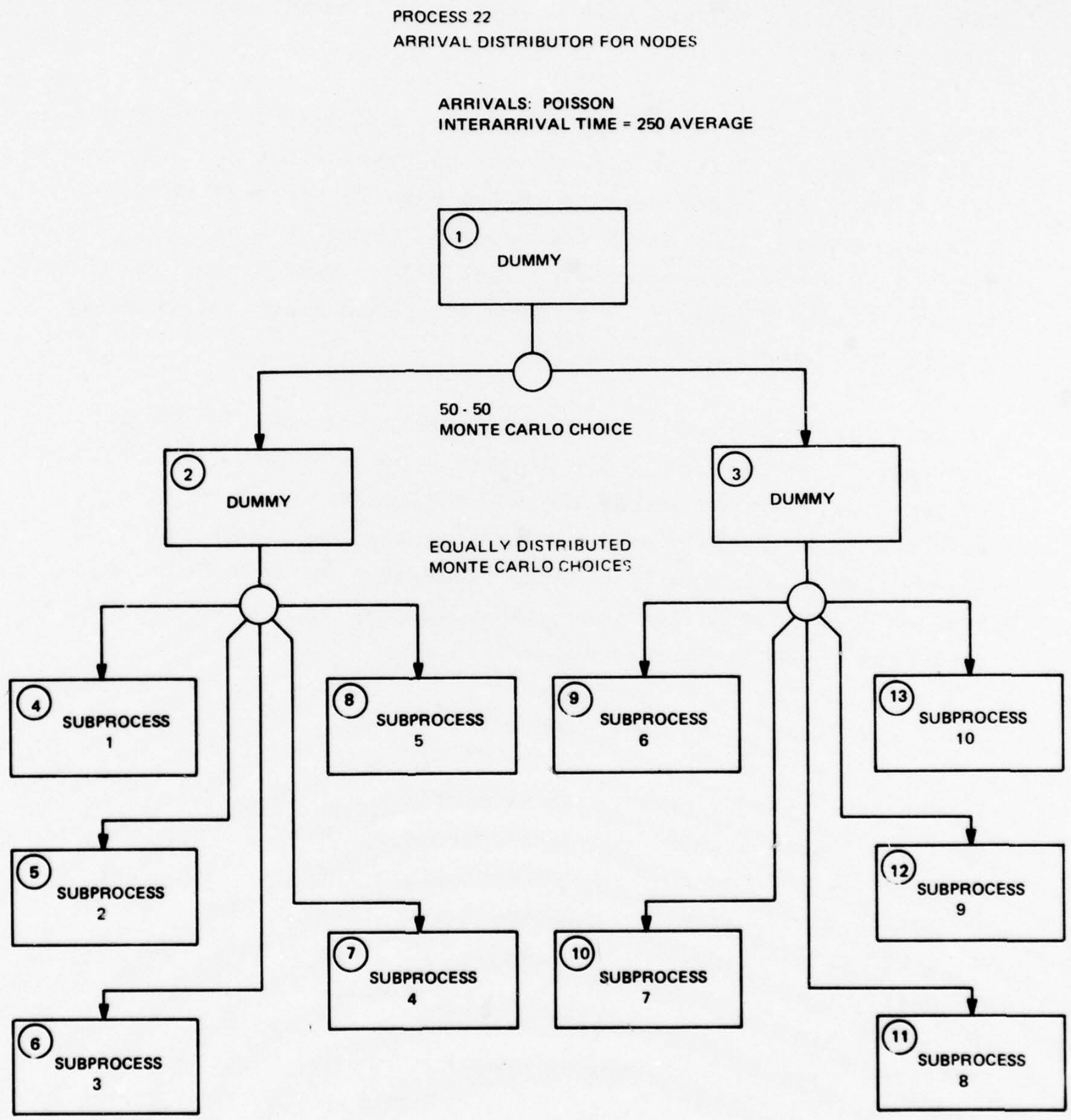


Figure 6-14
BOSS Process 22

Processes 11-20 (the node interface processes) show a combined median of 530, an overall mean of 671 and an average standard deviation of 424 time units. The ratio of standard deviation/mean is 0.632.

The queue statistics for unit types 11 through 20 (the node interfaces) show queue lengths averaging 0.08 to 0.13 with an overall average of 0.099. This compares favorably with the queue length of 0.105 shown in table 6-9 predicted by queuing theory. The sum of the utilizations for unit types 31 through 40 is 0.866 which compares well with the utilization factor of 0.856 in table 6-9. The sum of the utilizations for unit types 11 through 20 is 0.273. The value of ρ_N in table 6-7 opposite a load factor of 0.856 is 0.32 which is somewhat higher than 0.273. Considering the approximation of the queuing theory approach, the agreement is considered good. One other source of agreement is the ratio of standard deviation to mean for the interface time. The simulated value is 0.63, the calculated value is 0.67.

6.5.2.2.3 Bus Model

Processes 1-10 of the bus model are similar to processes 1-10 of the loop model (see figure 6-15). The differences involve the setting of "registers". Registers in BOSS are used to hold and change values of variables. At the completion of task 1, two registers are changed. R(N) is set to the value N indicating that the node N interface is busy. R(100) is incremented by 1 to indicate the number of interfaces busy. At the completion of any process N, R(100) is decremented by 1 and R(N) is returned to zero.

Processes 11-20 are the subprocesses called by the two tasks 2 of processes 1-10. Each is a single task that ends only when forced to end by process 22 which will be explained later.

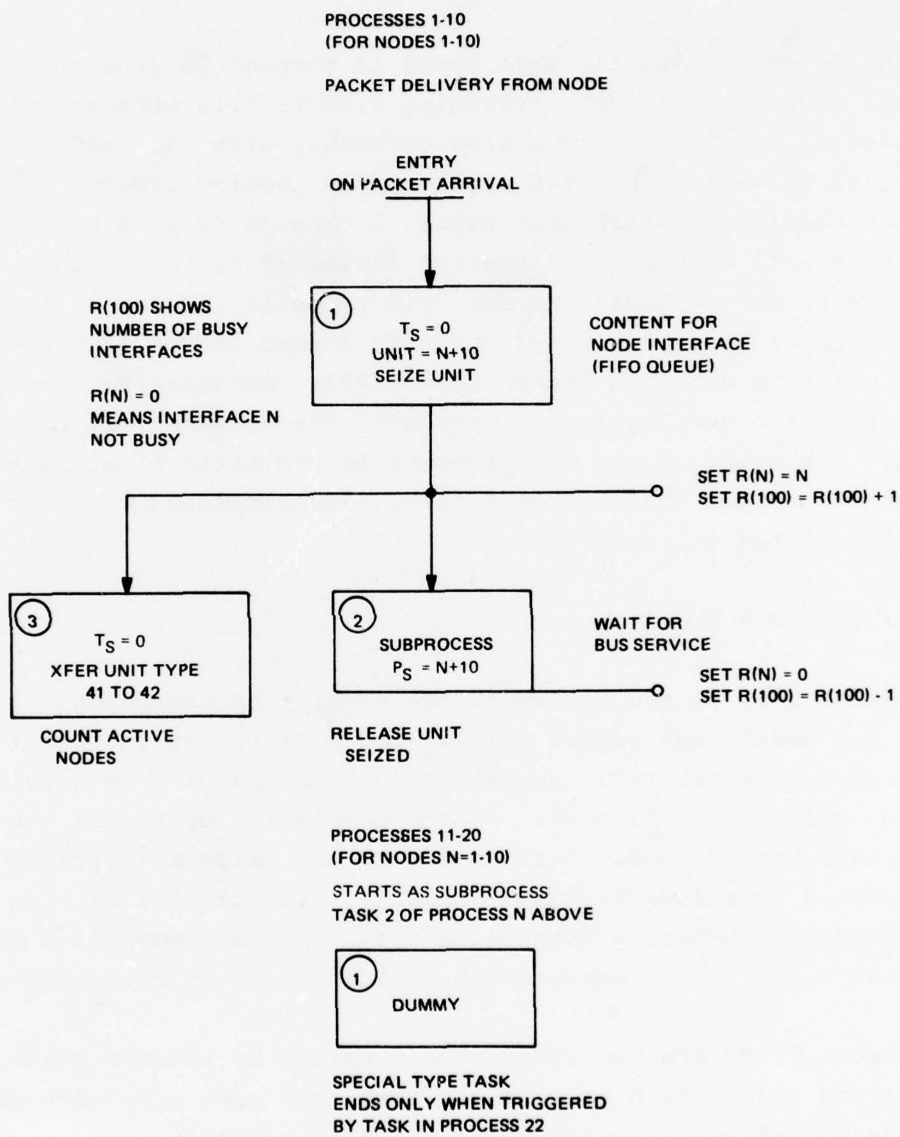


Figure 6-15

BOSS Processes 1-20 Bus Simulation

Process 21 is the heart of the simulation. It is the bus service process (see figure 6-16). Its start time is at 1 time unit and it is ever-running. Task 1 uses 1 time unit and a unit of type 42. If no units of type 42 are present, no processes 1-10 are active and the task queues on unit type 42. This saves considerable computer time during lulls. Task 2 is a scanner task that scans registers R(1) through R(10) and selects one of the registers that has a value greater than zero. The selection is random. In effect, one of the active node interfaces is selected at random. The register number is stored in R(13).

Task 4 is an IF task. If R(100)=0 (no busy nodes) an exit to task 3 occurs. This should never happen. It is put in the model only as insurance against the model hanging. If R(100)=1, task 5 (subprocess 22) is activated. If R(100) 1, task 6 is activated. This task determines whether a collision occurs as the result of two nodes getting the bus simultaneously.

Task 6 selects task 7, task 8 or task 9 according to whether R(101) contains a value of 2, 3 or more. If R(100)=2 then task 7 is chosen which selects a collision 4.5% of the time; if R(100)=3 then task 8 selects a collision 5.5% of the time else task 9 selects a collision 6.3% of the time. These values were taken from ETHERNET. If a collision occurs, (R(101) is incremented by 1 and the exit to task 3 is taken for retry. If a collision does not occur, task 11 (subprocess 22) is started.

Task 11 and 5 both call subprocess 22 when the subprocess ends, task 11 or task 5 ends. Exit is then made to task 3. Task 3 has a time of 1 time unit using the unit of type 3 that represents the bus. Feedback to task 1 then takes place and the whole procedure repeats.

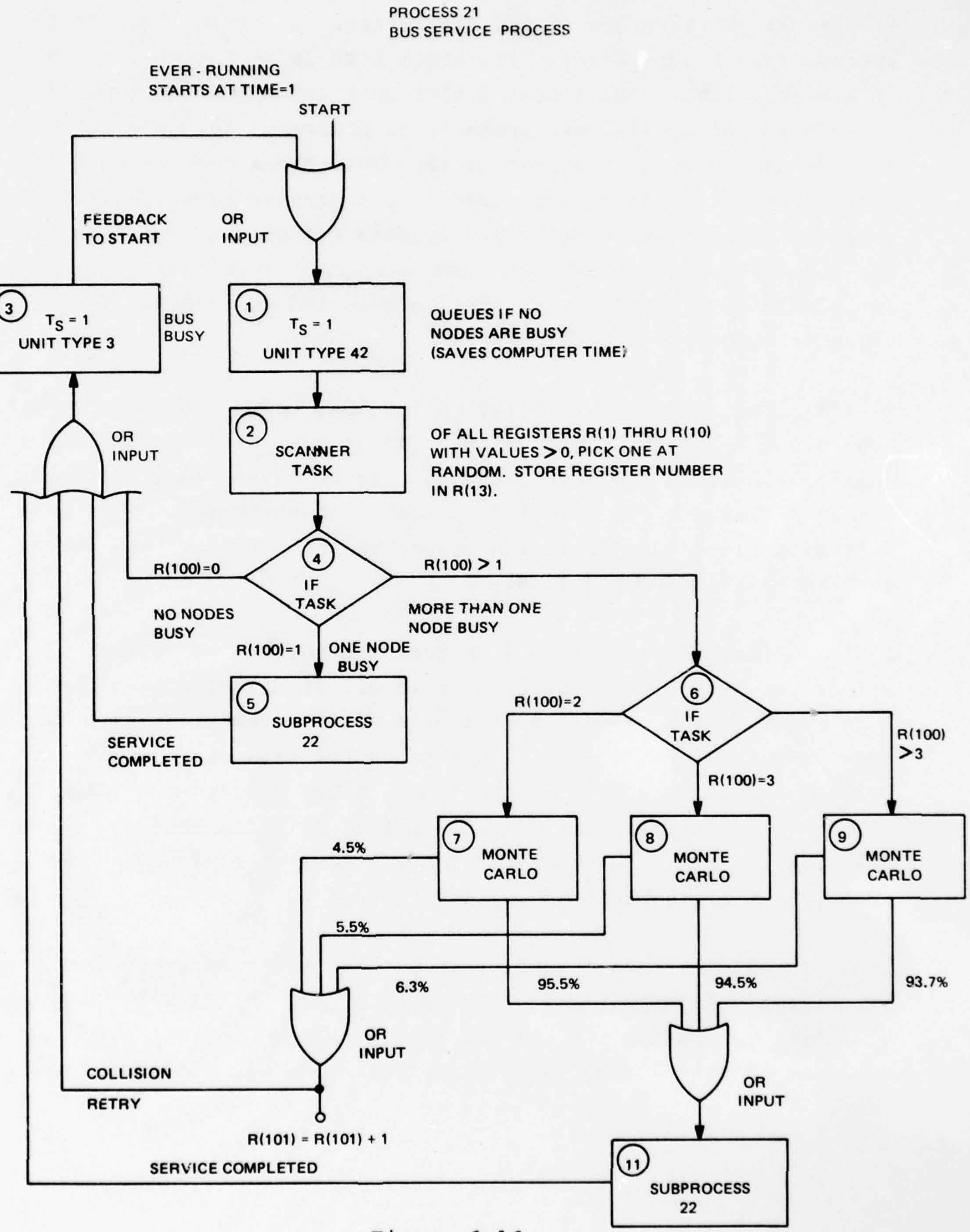


Figure 6-16

BOSS Process 21 Bus Simulation

Process 22 represents the bus service activity. It also causes the ending of one of the subprocesses 11-20 according to the value in R(13). Refer to Figure 6-17.

Process 22 has two start tasks (1 and 14). Task 14 is used only to keep the number of units in type 42 accurate by transferring one to unit type 41. Task 1 is the bus service task. It uses up 208 time units and the unit of type 3 busy. At its completion, it acts as a switch according to the value of R(13).

If R(13) is 1, 2, 3, 4 or 5 (representing processor 11, 12, 13, 14 or 15 respectively) then task 3, 4, 5, 6 or 7 is chosen which causes the ending of task 1 in process 11, 12, 13, 14 or 15. If R(13) is greater than 5, task 2 is chosen which causes R(13) to be decremented by 6. Task 8 is a switch task that causes task 9, 10, 11, 12 or 13 to be chosen which ends task 1 of process 16, 17, 18, 19 or 20. Thus, task 1 of process N+10 is ended as the result of R(13) having the value N.

Process 23 is identical to process 22 (Figure 6-14) of the loop simulation and has the identical purpose.

6.5.2.2.4 Results of Bus Simulation

Processes 1-10 show numbers of completions varying from 970 to 1070 giving a total of 10152 completions and a standard deviation of 32 which is close to the expected value of 31. The total of 10152 has a deviation of 152 which is not unreasonably different from the expected standard deviation of 100. Thus, the actual value of completions is about as expected and is substantially (within 0.27%) the same as that of the loop. Process 23, the conglomerate of all processes 1-10 shows a completion histogram with a minimum of 208 time units (as expected), a median of 484 time units, a mean of 927 time units and a standard deviation of 1347 time units. The ratio of standard deviation to mean is 1.453.

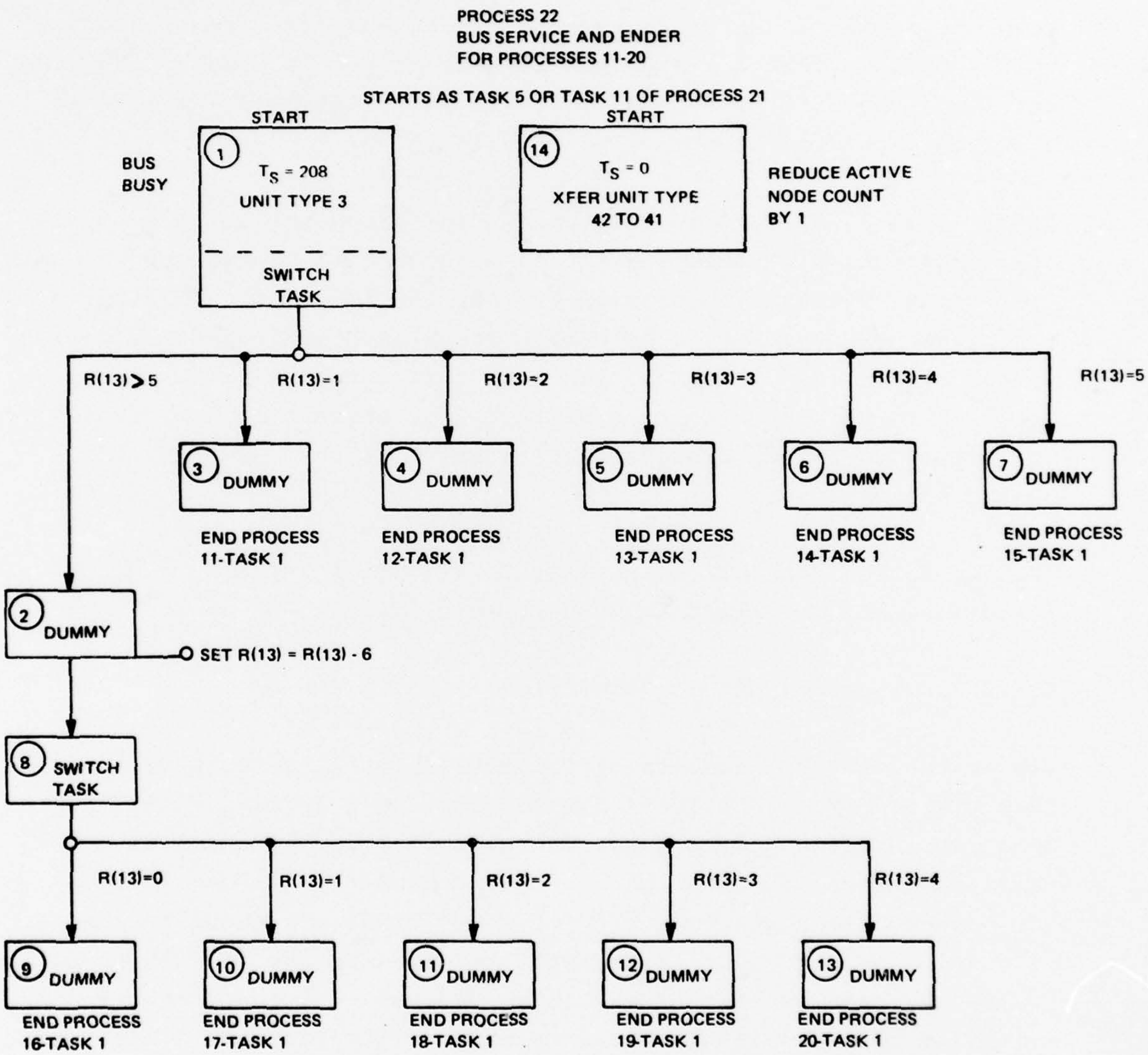


Figure 6-17
BOSS Process 22 Bus Simulation

Processes 11-20 (the node interface processes) show a combined median of 400, a combined mean of 620 and an average standard deviation of 674. The ratio of standard deviation to mean is 1.087 which is not far different from the expected ratio of 1.

The register status shows R(101) to have the value 415 which means that the collision rate was 4.1%.

The queue statistics for unit types 11 through 20 show queue lengths averaging 0.08 to 0.15 with an overall average of 0.123 which agrees well with the value of 0.130 shown in Table 6-9 predicted by queuing theory. The utilization for unit type 3 is 0.849 which is not far different from the utilization factor of 0.844 in table 6-9. The value of ρ_N in table 6-7 opposite the load factor of 0.84 is 0.292 which is somewhat higher than the average utilization for unit types 1 through 10 of 0.253. The agreement between simulated results and results from queuing theory is considered good.

6.5.2.3 Comparison for Throughput

The loop protocols of the Pierce and Reames types permit maximum throughputs with $\lambda T > 1$. The write token protocols WT-1 and WT-2 permit a $\lambda T=1$ as does the bus. Of the two WT protocols, the WT-2 has the greater queuing at the nodes and this has been demonstrated to be somewhat better than that of the bus. Thus, the loop is always superior to the bus.

The WT-2 protocol gives a guaranteed maximum orbit time of NT. This is true of none of the others including the bus. Thus in a 10 node, 2 megabit per second loop, service is guaranteed to each node in about 10 milliseconds in a WT-2 loop.

6.5.2.4 Comparison for Reliability

A serial bus is inherently free from vulnerability to most types of single node failures. A type of single node failure to which it is vulnerable is the case where a node puts noise on the bus. In such a case there is no way to tell which node is the offender. All nodes receive the offending noise essentially simultaneously. Even an extra bus is no guarantee of safety since the offender could apply noise to both busses.

Single node failures on a loop of a type that interrupt the loop cause loop failure. The addition of an auxiliary loop that counter-rotates with respect to the main loop may be used to overcome this disadvantage at little increase in cost through the use of "loop-back" or "wraparound" as described in Appendix B. Wraparound permits the isolation of any node or contiguous group of nodes from the rest of the nodes in the loop.

This also permits the isolation of a noise-producing node. Detection of such nodes is also relatively simple because of the sequential nature of the passage of information. For example, in a WT-1 or WT-2 loop where write tokens are regenerated after time-out, the node just downstream of the noisy node will be the only node that never receives a write-token. It can perform its own wrap-around and signal the node upstream of the noisy node to do the same. This will isolate the noisy node.

Another method that more generally applies is the use of framing character detectors. A noisy node will generally produce fewer framing characters than normal or else much too many. If any node detects such a condition over a period of time, it intercepts the bit stream and acts as a controller to search out the noisy node.

The bus can detect the noise condition but can do little to correct it automatically.

6.5.2.5 Comparison for Security

If noise in the form of hostile insertion is being applied to the loop or bus at some point (such as a gateway or some input to a physical part of the loop or bus) either to saturate the system or to saturate a node, the source can be isolated by loop-back in the case of a loop. It cannot easily be isolated in the case of the bus.

6.5.2.6 Comparison for Reach

In the case of a large geographically distributed loop or bus, the loop can be larger than a bus because each node is an amplifier/repeater and the bus is passive. Repeaters can be added to the bus to increase its reach, but the addition of such repeaters makes the bus more vulnerable to breakdown. Furthermore, when the reach of the bus is large, propagation delays increase the loss of time to detect simultaneous transmissions. On the loop, no such function exists.

6.5.2.7 Comparison for Flexibility

There are three types of loops, the write-token, Pierce and Reames types each with its own characteristics. The three types provide a choice of characteristics. The bus has only one structure available. Thus, if high throughput is required (λT higher than 1) then the Pierce or Reames loops should be chosen. This is particularly true where a large number of nodes is used. If an inherently non-blocking protocol is required with a guaranteed service time, then a write-token loop should be used. One case where the WT-2 loop is very advantageous, for example, is where flash messages are involved. Such flash messages need wait only the maximum orbit time which is usually quite short.

Another form of flexibility is in the flexibility of transmission medium. Fiber optic techniques of transmission for example are quite applicable to loops. The bus structure requires a tappable medium so that the transmission on the bus is not disturbed significantly by the tap. Thus, fiber optic techniques are not easily adaptable to the bus.

6.5.3 Sensitivity Parameter Variation

6.5.3.1 Parameters Considered

This section investigates the candidate designs with respect to the sensitivity analysis criteria of the proposal. The parameters investigated are: cost, size, adaptability/expandability, reliability, maintainability, performance, survivability, communications security, effectiveness when applied to DCS Concept of Operations, and implementation risk. Some of the above criteria can only be compared in qualitative terms, others can be compared in quantitative terms as evidenced by the Life Cycle Costing and Simulation analyses described above.

6.5.3.1.1 Cost Procurement/Life Cycle Cost

Life-cycle cost comparisons are given in Section 6.5.1. They are based on a similar strategy of spares warehousing and are calculated for 1000 MAS units total over a ten year period of acquisition. The MAS on which they are based is an estimate of what an actual MAS will be, including data acquisition modules but not included the test equipment from which these modules obtain the data.

The use of TI TM 9900's for the microcomputer modules will cost less than if DEC LSI-11 microcomputers are used. While the costs for the loop and the serial interface systems will be essentially the same once developed, the increment of cost for developing a serial LIU will be considerable both in terms of time and money.

6.5.3.1.2 Size

Regarding size, the loop and serial bus architectures will be approximately the same. Feasibility development models using either architecture should fit within a cabinet roughly the size of the ESMD loop 4 except that the program development unit will be external to the cabinet and housed in a small cabinet of its own. The TI microcomputer card is slightly smaller than the LSI-11 microcomputer card.

6.5.3.1.3 Adaptability/Expandability

The loop and serial bus architectures are equally expandable and adaptable. New modules and functions may be added up to the functional capability of the loop or buss. When that capability is exceeded, it may be enhanced through the use of more than one loop or bus as required with gateways connecting between the loops or busses. The various loop protocols that are available (WT-1, WT-2, Pierce, Reames) make the loop more adaptable to various applications than the bus. Loop protocols may be changed as the operating environment changes (e.g., low traffic vs. high traffic conditions, flash message capability).

6.5.3.1.4 Reliability

The loop with its automatic loop-back tends to make it highly reliable except that the clock must be made redundant to ensure that a single failure (even though the MTBF for a clock is very high) will not shut the system down. The serial bus is inherently very reliable since the failure of individual BIU's does not halt the

operation of the bus. The double loop is significantly more reliable than the bus since there is greater capability for fault-isolation procedures (e.g., using WT passing detection schemes for isolating a node that has failed in a constant write state).

The LSI-11 module is slightly more reliable than the TMS 9900 module due to the memory reliability. The LSI-11 uses dynamic random access meory (RAM) chips and the TMS 9900 uses static RAM chips. The dynamic RAM's are more reliable than the static RAM's, however the time required for refreshing degrades performance. The industry tends to be gravitating in the direction of static RAM's. On a system basis, since more LSI-11 modules are required, the system reliability is better for the TMS 9900 configuration.

Additional discussion on reliability is found in Sections 6.5.1 and 6.5.2.

6.5.3.1.5 Maintainability

The loop and bus are both extremely maintainable due to their high degree of modularity. Both the LSI-11 and TMS 9900 are contained on a single card. Memory will be contained on one or two cards. The LIU is contained on one card. It is anticipated that a BIU would be contained on one card.

The maintenance philosophy will be to isolate the bad card and replace it from a set of spares. The bad card can then be tested to isolate bad chips. A set of diagnostic routines exists for the loop; the routines would have to be developed for the bus. The automatic bad node isolation capability found in the loop allows hot-card replacement; (i.e., the system need not be powered down or affected while faulty cards are replaced. It is anticipated that a similar capability could be developed for the bus.

6.5.3.1.6 Performance

The loop performance is superior to the bus performance as discussed in detail in Section 6.5.2. The loop has smaller queues at high input traffic rates than the bus. This is due to the large probability of bus collision at high traffic rates. The worst, in terms of throughput, loop protocol (WT-2) is better than the bus; other loop protocols (WT-1, Pierce, Reames) are considerably better than the bus in throughput. The interprocessor network acquisition time is bounded for the write token (WT-2) loop; it is not for the bus. This means that a FLASH message can be guaranteed to be delivered within a maximum time period for the loop but not for the bus. This is important for delivering high priority alarm reports generated by a module which detects a fault conditon. The alarm would have to be sent from the originating module to the OCRI, DBMS, and SSCI modules in a short guaranteed time via the interprocessor network.

The TMS 9900 is considered superior in performance to the LSI-11 as discussed in Chapter 4.

6.5.3.1.7 Survivability

Survivability should be excellent for the loop. Any module that fails (except the data base module, DBMS, and clock which could both be duplicated) can be supplemented by a standby module. The loop itself is self-healing owing to the automatic loop-back feature. It is anticipated that the survivability for the serial bus should also be very good. The I^2L version of the TMS 9900 (i.e., the SBP 9900) has excellent survivability as compared to the NMOS technology candidates. A militarized version of the LSI-11 is available from Norden.

6.5.3.1.8 Communications Security

All of the selected architectures have equivalent communications security. If one assumes that each node and sector are secure in themselves and that the equipment at the station level is physically secure, then the problem of communications security becomes one of link encryption for the 2400 baud lines between stations and nodes, nodes and sectors. Once the loop or serial bus with the attendant LIU's are physically secure against tampering, then they are secure. Only the microcomputers are changable from the outside and then only as the result of messages. If the link encryption is sufficiently good to ensure this, then the system is secure provided their programs are correct.

6.5.3.1.9 Effectiveness of Designs to DCS Concept of Operation

One of the main ideas behind the modular system control development model program has been the development of a modular approach to the lower three echelons of the DCS tree; namely the sector, the node and the station levels. The implementation of the ACOC and DCAOC levels were considered beyond the purview of this study.

The station, the node and the sector level configurations are implemented using different subsets of the same set of modules in each case. The only differences in the modules themselves is one of size. Thus, if LSI-11 microcomputer modules were used at the station level then PDP-11 modules might be used at the node and sector levels; if TI 990/4 modules were used at the station level, then TI 990/10 modules might be used at the node and sector levels.

As regards the issues of survivability, reliability, modularity, adaptability, expandability and issues of that sort, there seems little doubt that the modular approach is superior to the centralized approach. Since there need be no more modules than necessary in the modular approach, whereas the centralized approach would require a machine large enough and fast enough for the peak situation, it would seem that the modular approach would result in a smaller,

lower cost system as well. Furthermore, the simplicity of the module hardware would seem to indicate that a simpler spares policy and a lower level of maintenance knowledge would be required.

6.5.3.1.10 Implementation Risk

The loop architecture has less implementation risk than the bus architecture. This is because the loop interface unit (LIU) has already been developed at ADO for the double loop while the bus interface unit (BIU) has not. In addition Burroughs has considerable experience (cf. Appendix B) in implementing loop communication networks including diagnostic and fault detection and isolation software. System operation experience has provided the inputs for these sensitivity analysis criteria for the loop, while the inputs for the bus are in some cases based upon anticipated operation. There seems to be little difference in implementation risk for the two microcomputer candidates.

6.5.3.2 Parameter Variation

This Section will vary each of the above criteria to determine the effect on the other criteria. The criteria discussed below which were chosen to be varied are module cost, system size (i.e., number of modules), module reliability, and module performance. The other criteria either are not easily varied (e.g., effectiveness of proposed designs when applied to the DCA Concept of Operations), or had little effect on the other criteria when varied (e.g., increased maintainability only reduces the maintenance cost in the Life Cycle Cost). The results of the parameter variation are summarized in Tables 6-10 to 6-14. As the considered criterion is increased the other criteria are indicated to increase (I) or decrease (D). The criterion is indicated to be affected more in a bus (B) or loop (L) Implementation.

6.5.3.2.1 Cost

Table 6-10 indicates the criteria variance with respect to an increase in module cost. As the module cost increases, adaptability/flexibility decreases since less equipment would be supplied with a fixed price system. This decrease affects the bus architecture more since the multiple protocol feature of the loop gives it more adaptability/flexibility than the bus. Maintainability decreases as module cost increases since the cost for a set of spares is increased, and a module costs more to repair. As module cost increases the performance decreases since less equipment would be supplied with a fixed price system. The above variation could be compared to the increased module cost of the LSI-11 over the TMS 9900. In that case, system size would also increase slightly since the LSI-11 microcomputer card is slightly larger than the TMS9900 microcomputer card.

Table 6-10
Module Cost Increase I

	I/D	L/B
Size	-	-
Adaptability/Flexibility	D	B
Reliability	-	-
Maintainability	D	-
Performance	D	-
Survivability	-	-
Comm. Sec.	-	-
Conc. of Oper.	-	-

6.5.3.2.2 Size

As the system size in terms of number of modules required increases, all criteria increase except for communications security and concept of operation which are not affected as indicated in Table 6-11. The increase in adaptability/flexibility, reliability and performance is more significant in the loop than the bus since the loop is superior to the bus with respect to these criteria.

Table 6-11
System Size (# Modules) Increase I

	I/D	Affect Loop/Bus More
Cost	I	-
Adaptability/Flexibility	I	L
Reliability	I	L
Maintainability	I	-
Performance	I	L
Survivability	I	-
Comm. Sec.	-	-
Conc. of Oper.	-	-

The sensitivity of the number of modules as a function of microcomputer processing speed lies primarily with the VSQC modules (in particular, the speed of FFT performance). Assuming the present assessment to be correct, the LSI-11 module can perform quality control of 360 voice channels per hour whereas the TI 990 can perform the same function on 550 channels per hour. Table 6-12 shows the results of a sensitivity analysis in terms of total modules required for the feasibility development model based on a speed variation of 50 per cent plus or minus. This estimate is based on 1000 channels of voice.

Table 6-12
Sensitivity Analysis of Total Modules
Versus Processing Speed

Speed Variation	Number of Modules Required TI 990	LSI-11
-50%	8	10
-25%	7	8
NOMINAL	6	7
25%	6	7
50%	5*	6

* It may be that VSQC and DSQC modules should be mapped on separate microcomputer modules so that 6 modules would be minimum.

There is a concomitant increase in size and power for each added module. Communications security does not appear to be involved.

6.5.3.2.3 Reliability

Module reliability variance is summarized in Table 6-13. As module reliability increases, the repair cost decreases, maintainability increases, and performance increases due to less down time. This variation corresponds to the increased reliability of the LSI-11 module as compared to the TMS 9900 module due to dynamic RAM's being more reliabile than static RAM's. However, this analogy does not correspond to the actual FDM system design since more LSI-11 modules are required due to the superior performance of the TMS 9900; as a result the system reliability is superior for the TMS 9900 based system.

Table 6-13
Module Reliability I

	I/O	L/B
Cost	D	-
Size	-	-
Adaptability/Expandability	-	-
Maintainability	I	-
Performance	I	-
Survivability	-	-
Comm. Sec.	-	-
Conc. of Oper.	-	-

6.5.3.2.4 Performance

Table 6-14 summarizes module performance variance. As the module performance is increased the system cost and size are decreased since less modules are required. Also system adaptability/flexibility, maintainability, and reliability are increased since less modules are required. This increase is greater for the loop since it is more adaptable/flexible, reliable, and maintainable than the bus. This variation corresponds to the TMS9900 which has increased performance as compared to the LSI-11.

Table 6-14
Module Performance Increase

	I/D	L/B
Cost	D	-
Size	D	-
Adaptability/Flexibility	I	L
Reliability	I	L
Maintainability	I	L
Survivability	-	-
Comm. Sec.	-	-
Conc. of Oper.	-	-

6.5.3.2.5 Summary

The sensitivity criteria summary is given in Table 6-15. The loop (L) is superior for least development cost, adaptability/expandability, reliability, maintainability, performance, and low implementation risk. The TMS 9900 (T) is superior for least hardware cost, smallest size, performance, and survivability (the I^2L version). The LSI-11 (D) module is more reliable and maintainable, however since more are required for the system, the system reliability and maintainability is less.

Table 6-15
Sensitivity Criteria Summary

	Superior Architecture	Superior Microcomputer
Least Development Cost	L	-
Least Hardware Cost	-	T
Smallest Size	-	T
Adaptability/Expandability	L	-
Reliability	L	D*
Maintainability	L	D*
Performance	L	T
Survivability	-	T(I^2L)
Comm. Sec.	-	-
Conc. of Oper.	-	-
Least Implementation Risk	L	-

*Even though the DEC microcomputer module is more reliable and maintainable than the TI microcomputer module, the TI system is more reliable and maintainable than the DEC system since more DEC modules are required.

6.6 Conclusions and Recommendations

The recommended feasibility development model architecture based upon the above analyses is the loop architecture of Figure 6-1 using the TMS 9900 microcomputer module with the hardware requirements of Table 6-1. Six of the seven required microcomputers will be model TM990/100M with the seventh being a TI990/4 which is part of the program development unit. Each microcomputer wil have an associated 64K bytes of memory (model TM 990/202 memory modules). The actual deliverables for the recommended loop-TI architecture are given in Table 6-16. A suggested Option A at additional cost would be to use a TI 990/10 minicomputer development system which would perform the functions of PDU, DBMS, and OCRI requiring one less node on the loop. Option A would provide DCEC with a more powerful simulation facility due to the additional minicomputer memory and disk capacity.

Table 6-16
Loop-TI Recommended Architecture

Module	Quantity
Microcomputer	6
32K word memory	6
LIU	9
PDU	1
10K word memory	1
FORTRAN compiler	1
Card Racks	6
B776 cabinet with power supply	1
TI Silent 743	1

Option A: Substitute a TI990/10 minicomputer development system for the PDU, 10K word memory, FORTRAN compiler and one microcomputer, 32K word memory, LIU, and card rack.

Option B using a loop configuration with DEC LSI-11 microcomputers is given in Table 6-17. Note that an additional module is required as compared to the TI recommended architecture in order to satisfy the SOW. A serial bus implementation would require the same hardware of Tables 6-16, 6-17 except that a Bus Interface Unit (BIU) rather than a Loop Interface Unit (LIU) would be required. It is estimated that the development of the BIU would require 3-4 months of additional time and 6-8 man-months of additional effort compared with the loop implementation.

Table 6-17
Loop-DEC Architecture (Option B)

Module	Quantity
Microcomputer	7
32K word memory	7
LIU	10
PDU	1
32K word memory	1
Card Racks	7
B776 cabinet with power supply	1
TI Silent 743	1

7. USER LANGUAGE DEFINITION

7.1 Introduction

The OCRI column function discussed in Chapters 2 and 3 provides the interface between man and machine. The Feasibility Development Model (FDM) operator or user language must provide DCA personnel with the ability to control a SYSCON simulation facility. In this study the ATEC and TCCF operator languages were investigated in order to identify SYSCON functions. It is desirable that the FDM operator language be similar to the ESM operator language (program USRLNG on the PDP-11's) so that the entire SYSCON simulation facility would have a similar user interface. This approach would minimize the training time required for DCA personnel to become familiar with the system.

The ESM user language consists of five modes of operation (CRT to CRT, System Inquiry, System Control, File Access, Card Format). Each mode of operation is contained within one or more FORTRAN subroutines. Modules can easily be updated, added or deleted. The language uses a "menu-selection" type of operation which implicitly instructs the inexperienced operator on operating the system.

For an actual SYSCON station, both a "menu-selection" type of operation for inexperienced operators and a "command and control language" type of operation for experienced operators is desirable. For the FDM simulation facility however, the "menu-selection" type of operator language is more desirable since demonstrations will often be performed for inexperienced personnel (e.g., HSF personnel not familiar with SYSCON applications). A limited amount of commands (e.g., "DS" for logging out anywhere in the dialogue) will be useful. As operator experience increases, DCA personnel can add commands by modifying the user language application program using the FDM software development facility.

7.2 Startup and Loading Procedures

The following startup and loading procedure is suggested for the FDM:

- There will be a PROM loading chip (256 x 8) associated with each FDM microcomputer. The loading software contained in these chips will be of two types: type 1 which receives load data from the interprocessor communication network, and a single chip of type 2 which receives load data from the Program Development Unit (PDU).

- Object files are stored on PDU disk with a loader application program to be supplied written in FORTRAN. The loader program runs on the PDU and prompts for a normal system load (e.g., as in Figures 6-1 or 6-2, 7TI or 8 LSI-11 object files), or an individual node load. Positive response after each node is loaded is displayed on the CRT.

- A master load switch allows microcmputers to fetch instructions from the PROM laoder chip rather than RAM.

- Type 1 PROM loader programs set up a load read address and EOF recognition address in the address comparison memory (ACM). When a packet is read the information bytes are written into sequential locations of RAM starting at location 0. When an EOF indicator is detected an ACK message is sent back to the PDU connected node. The ACM is then modified so as not to recognize any addresses and the microprocessor performs a do nothing loop.

- The type 2 PROM loader program loads its RAM with instructions if the packet address from the PDU equals its load address. Otherwise, it sends packets out on the interprocessor communications network. After sending an EOF indicator, it waits for a positive acknowledgement which is sent to the PDU.

It is expected that the system parameters will be supplied as default values within the source code (FORTRAN) of the column functions. When the system is in the running state, certain parameters can also be changed in real time through the issuance of parametric change messages as required. Certainly such parameters as connectivity parameters and threshold levels as well as routing tables should be changeable in real time through the use of messages sent from module to module and messages sent from other simulated echelons. The communications system, column function programs, and user language program will be designed to provide such a capability.

7.3 Modes of Operation

The recommended modes of operation for the FDM User Language are described below:

Mode 1: CRT-to-CRT. In this mode a user CRT may send messages to another CRT. For example, an FDM operator using the TI 743 terminal can send a message to an ESM TD802 terminal. This mode is human to human dialogue and messages are sent in free text format (up to 240 bytes of information). This mode of operation can be used to simulate communication between system controllers at different sites.

Mode 2: System Inquiry. This mode of operation is used to obtain information about the FDM itself. A typical display would include node designators, functional addresses, logical identifier definitions, and default node column function assignments.

Mode 3: Module Update. This mode of operation would be used to change parameters in the software implemented column functions. The User Language program would generate special parameter change control messages that would be interpreted by the nodes.

Mode 4: File Access. This mode of operation provides the data base management language permitting records of files to be accessed, edited, added, and deleted.

Mode 5: Report. This mode of operation allows reports to be generated by the SYSCON controller. This may be used to simulate reports that are generated with computer assistance, and sent to higher levels of the SYSCON hierarchy.

Mode 6: Status. This mode of operation will provide the operator with status displays of site and equipment performance.

The above six modes of operation will be implemented as subroutines in a single application program or a set of application programs that run on the OCRI microcomputer depending on the memory requirements. In addition to the above modes of operation the system will operate in other modes of operation depending on utility programs that can be run on the Program Development Unit (PDU). This includes normal PDU operation when simulations are not being performed and the PDU CRT interfaces with the PDU operating system. The user would have access to the specific PDU utilities (e.g., compilers, editors), and the PDU along with its peripherals is capable of being used independent of the other FDM equipment. The data base definition language will be implemented on the PDU as one or more FORTRAN utilities that would generate a baseline set of files for the FDM data base (e.g., files which contain status information for OCRI displays). Microcomputers will be loaded by a loading utility which will run on the PDU.

Failure messages and alarms will be printed on the TI 743 terminal interrupting the User Language dialogue. These messages will inform the operator of both actual and simulated failures, and the messages will be given a flash priority in the network. CRT-to-CRT messages from the ESM will also interrupt the User Language dialogue.

7.4 Example Dialogue Description

The following Host-CRT dialogue is given as an example of a possible implementation of the FDM User Language. Flowcharts for the dialogue are given in Figure 7-1 to 7-7. Note that if the recommended configuration of Figure 6-1 or 6-2 is implemented, the CRT response will be typed on the TI 743 terminal, and the HOST response will be generated by the OCRI microcomputer. Displays obtained from files on mini disk will be supplied by the DBMS microcomputer. The dialogue may be terminated via logout at any time by entering "DS" at the terminal.

CRT: * BEGIN

HOST: THIS IS THE FDM (FEASIBILITY DEVELOPMENT MODEL)
ENTER USERCODE PLEASE

CRT: Usercode

HOST: ENTER PASSWORD PLEASE

CRT: Password

HOST: YOU ARE NOW LOGGED IN - (TO LOGOUT, ENTER "DS")
PLEASE SELECT ONE MODE OF OPERATION:
1. CRT-TO-CRT.
2. SYSTEM INQUIRY.
3. MODULE UPDATE.
4. FILE ACCESS.
5. REPORT.
6. STATUS.

CRT: 1 - 6

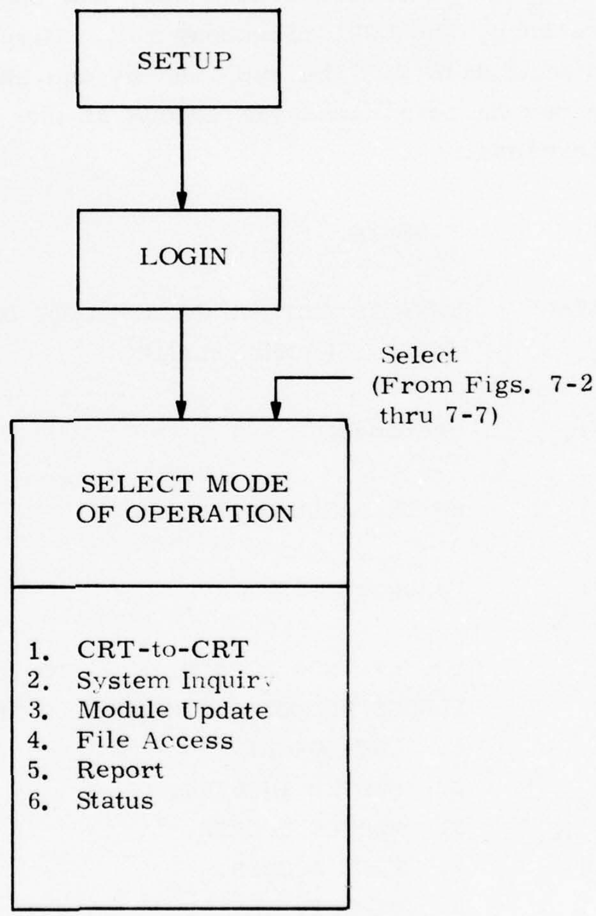


Figure 7-1
FDM User Language

7.4.1 Mode 1 CRT-to-CRT (See Figure 7-2)

HOST: ENTER DEST CRT NODE DESIGNATOR (ND) - 4 FOR LP#2, 8 FOR LP#3, 12 FOR LP#4, 24, 25 FOR FDM. IF NOT KNOWN ENTER "CRT"

CRT: 4, 8, 12, 24, 25 or CRT (if CRT HOST displays possible destinations)

HOST: PLEASE TYPE IN MESSAGE AND TRANSMIT

CRT: The message. (Enter on first 3 lines of CRT.)

HOST: PLEASE SELECT ONE MODE OF OPERATION:
1. NEW MESSAGE TO SAME CRT.
2. NEW MESSAGE TO ANOTHER CRT.
3. LOGOUT.
4. NEW MODE OF OPERATION.

CRT: 1 - 4

Dialogue now repeats as shown in Figure 7-2. If 3 is chosen, then

HOST: YOU ARE LOGGED OUT FROM FDM

7.4.2 Mode 2, System Inquiry (See Figure 7-3)

HOST: PLEASE SELECT TYPE OF SYSTEM INFORMATION:
1. NETWORK MODULE INFORMATION.
2. LID/FAD CONVERSION TABLE (LID's 1-100)
3. LID/FAD CONVERSION TABLE (LID's 101-254)
4. WORKPAGE PARAMETERS OF NODE.

*CRT: 1 - 4

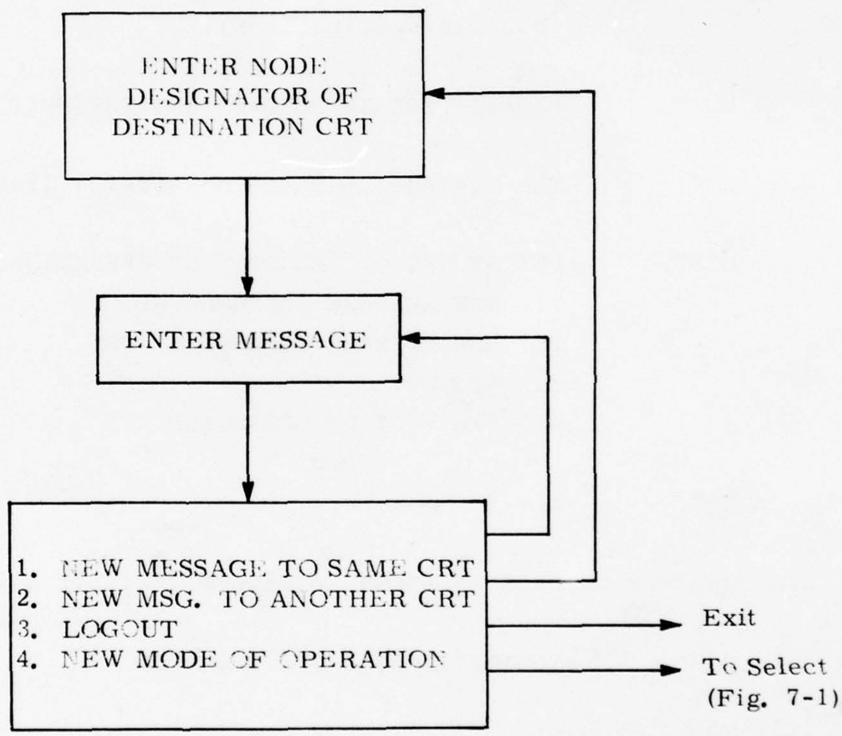


Figure 7-2
CRT-to-CRT Mode of Operation

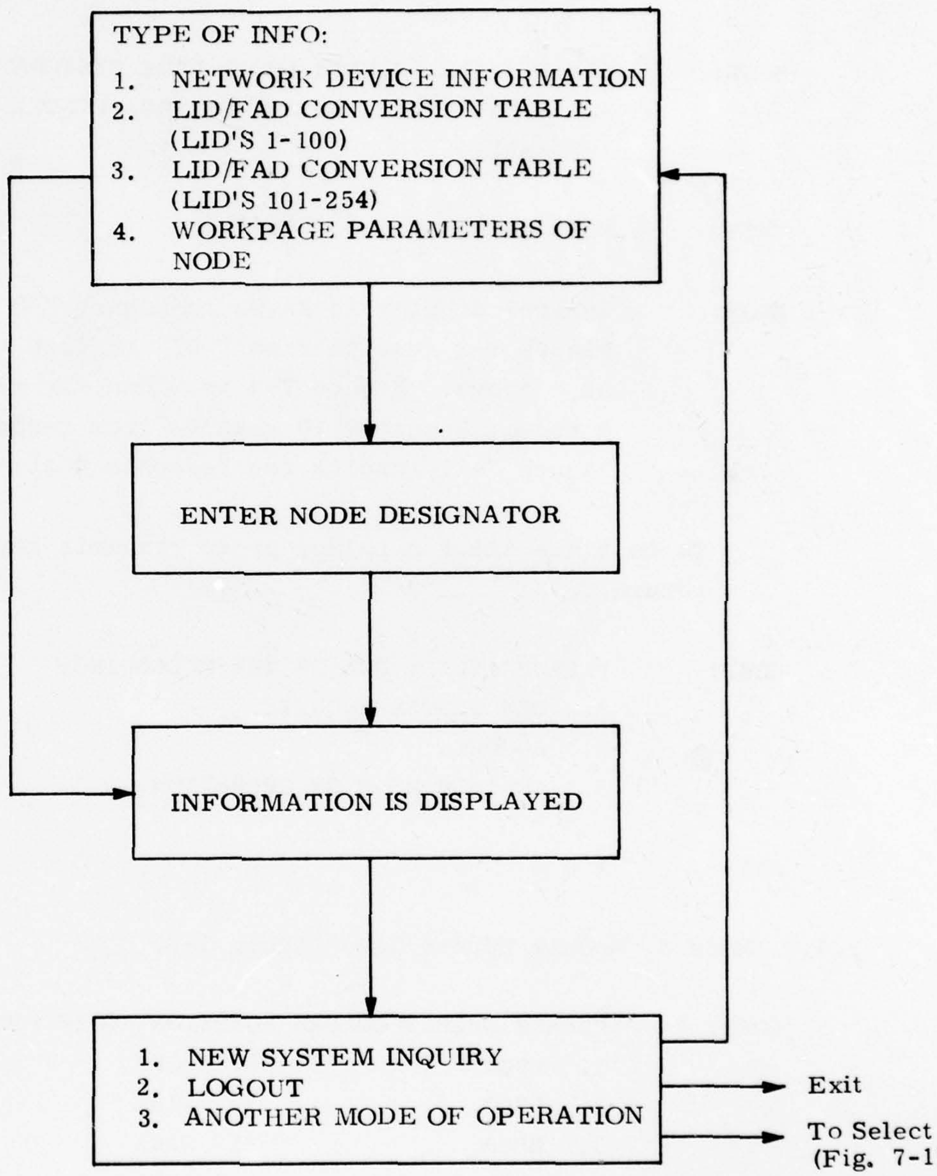


Figure 7-3
System Inquiry Mode of Operation

HOST (If 1): See Figure 7-8 for typical display.

HOST: (If 2 - 4): PLEASE ENTER NODE DESIGNATOR (ND). IF ND IS NOT KNOWN, ENTER NDI FOR NETWORK DEVICE INFORMATION.

CRT: Node designator or "NDI".

HOST: Typical display is shown in Figure 7-8 through 7-11. Figure 7-8 results from "NDI" or from response 1 at * above. Figure 7-9 is shown for response 2 at * above; Figure 7-10 results from response 3 at *; Figure 7-11 results for response 4 at *.

To continue after display, press transmit key (carriage return).

HOST: PLEASE SELECT ONE OF THE FOLLOWING:
1. NEW SYSTEM INQUIRY.
2. LOGOUT.
3. ANOTHER MODE OF OPERATION.

CRT: 1 - 3

7.4.3 Mode 3, Module Update (see Figure 7-4)

HOST: PLEASE SELECT COLUMN FUNCTION TO BE UPDATED

1. VSQC	6. OCRI
2. DSQC	7. FIAC
3. BBSA	8. SSCI
4. WBSA	9. DBMS
5. SDCA	

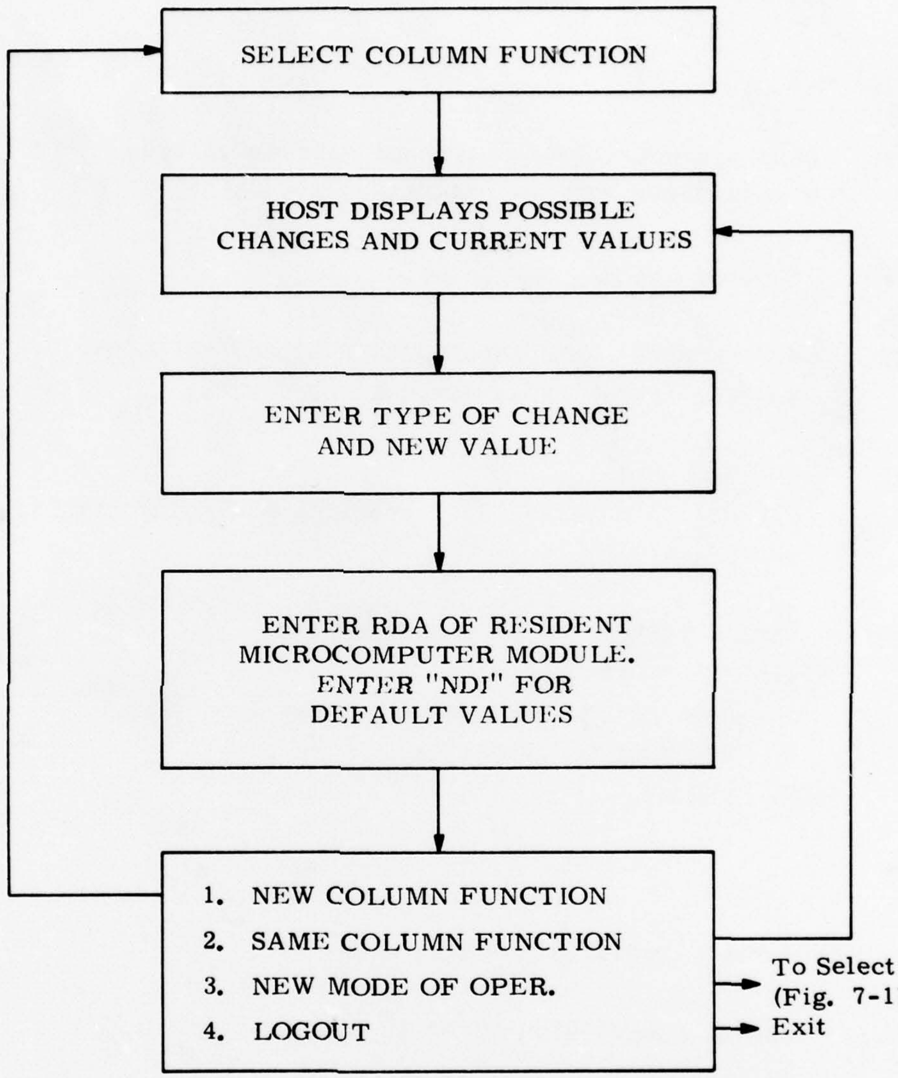


Figure 7-4

Module Update Mode of Operation

CRT: 1 - 9

HOST: Displays possible changes and current values
PLEASE ENTER TYPE OF CHANGE AND NEW VALUE

CRT: Type of Change, New Value.

HOST: PLEASE ENTER READ ADDRESS OR RESIDENT MICROCOMPUTER MODULE. ENTER "NDI" FOR DEFAULT VALUES.

CRT: 1 - 9 or NDI
(If NDI is entered, HOST provides a display similar to Figure 7-8).

HOST: PLEASE SELECT ONE OF THE FOLLOWING:
1. NEW COLUMN FUNCTION
2. SAME COLUMN FUNCTION
3. NEW MODE OF OPERATION
4. LOGOUT

7.4.4 Mode 4 File Access (See Figure 7-5)

HOST: PLEASE SELECT FILE TO BE ACCESSED:
1. LOCATION FILE.
2. CIRCUIT DIRECTORY.
3. TURNK DIRECTORY.
.
.
.

CRT: 1 -

HOST: A RECORD OF THE FILE YOU HAVE SELECTED HAS THE FOLLOWING FORMAT:

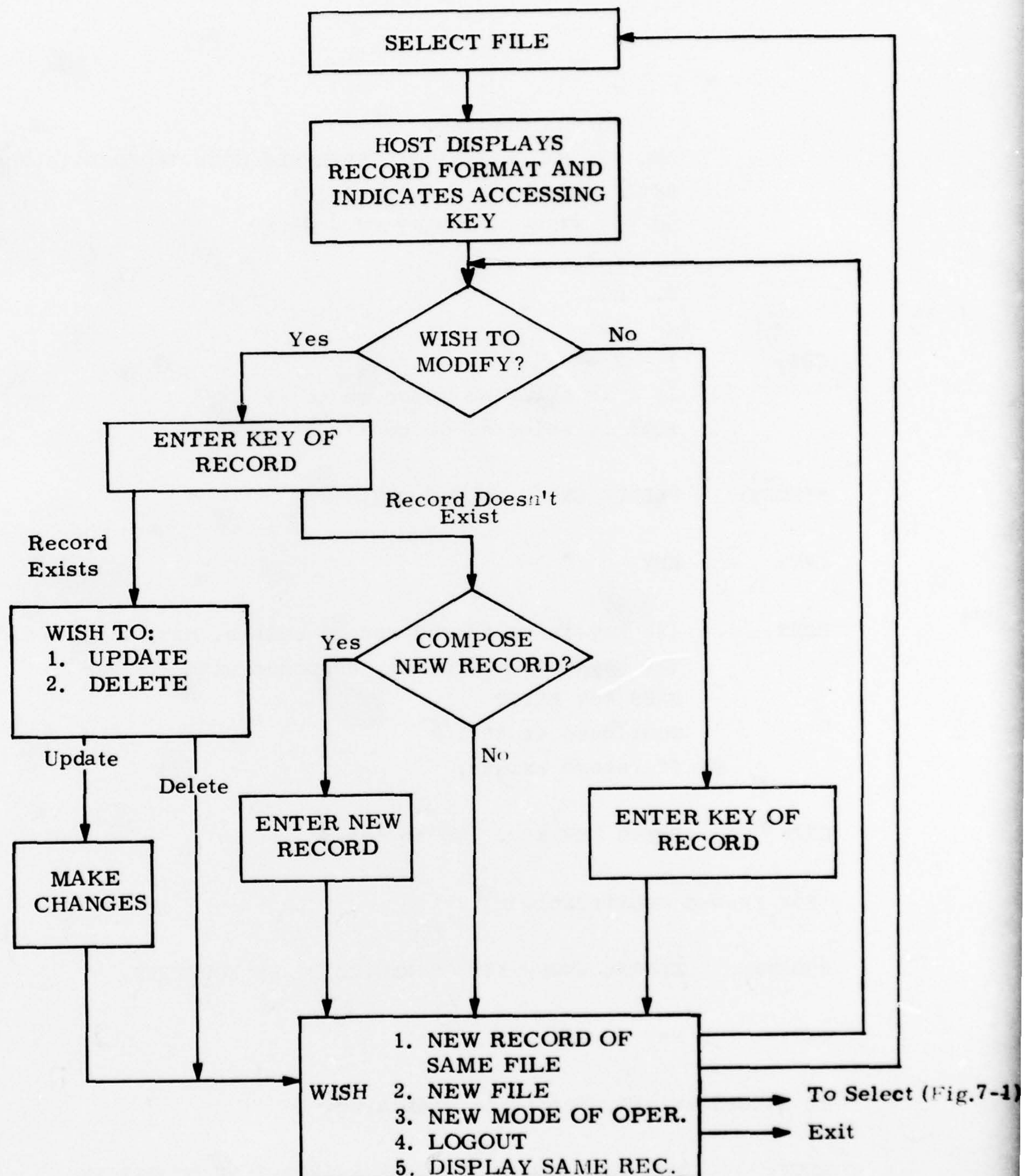


Figure 7-5

File Access Mode of Operation

Format Display.

THE KEY HAS A (Value) CHARACTER CODE IN FORM (ALPHA-BETIC, NUMERIC, ALPHANUMERIC)
DO YOU WISH TO MODIFY THIS FILE?
1. YES.
2. NO.

CRT: 1 - 2
If 1 is selected above go to *.
If 2 is selected go to **.

**HOST: PLEASE ENTER ACCESS KEY.

CRT: KEY

HOST: (If key is valid and record exists, displays record.)
(If key does not exist, responds with THE RECORD DOES NOT EXIST
continues at ***)
If record exists,

CRT: Press XMT key. Go to ***.

*For record modification

@HOST: PLEASE ENTER KEY OF RECORD TO BE MODIFIED.

CRT: Key

If record exists, record is displayed.

HOST: FOR THIS RECORD, PLEASE SELECT TYPE OF DESIRED CHANGE:
1. UPDATE.
2. DELETE.

CRT: 1 - 2. If 2 to to @@
If 1, record to be updated.

HOST: MAKE ANY CHANGES YOU WISH USING CRT KEYBOARD. WHEN CHANGES ARE COMPLETE, PRESS XMT KEY. ENTER UPDATED RECORD ON CRT. (Record is entered on CRT)

CRT: Edits and transmits. Go to @@.
If record does not exist,

HOST: THE RECORD DOES NOT EXIST. DO YOU WISH TO ADD A RECORD TO THE FILE?
1. YES
2. NO

CRT: 1 - 2. If 2 go to @@
If 1, record to be added

HOST: Give record format.
KEY IS (X) CHARACTERS OF TYPE (type).
ENTER THE RECORD ACCORDING TO THE ABOVE FORMAT.
WHEN RECORD IS COMPLETE, PRESS XMT KEY.
ENTER NEW RECORD ON CRT.
(Enter on CRT)

CRT: Edits and transmits.

@@HOST: MODIFICATION COMPLETE. (Record unlock occurs.)

***HOST: PLEASE SELECT ONE OF THE FOLLOWING:
1. NEW RECORD OF FILE
2. NEW FILE.
3. NEW MODE OF OPERATION.
4. LOGOUT.
5. DISPLAY SAME RECORD.

(Dialogue repeats as shown in Figure 7-5.)

7.4.5 Mode 5, Report (see Figure 7-6)

HOST: PLEASE SELECT TYPE OF REPORT TO BE GENERATED
1.
2.
.
.
.

CRT: 1 -

#HOST: Displays report format.
PLEASE ENTER REPORT PARAMETERS TO BE CHANGED.

CRT: Report parameters.

HOST: Displays report.
DO YOU WISH TO EDIT MORE?
1. YES
2. NO

CRT: 1 - 2
if 1 go to #
if 2,

HOST: ENTER DESTINATION NODE DESIGNATOR (1 for LP#1,
5 for LP#2, 12 for LP#4).

CRT: 1, 5, 12

HOST: PLEASE SELECT ONE OF THE FOLLOWING:
1. NEW REPORT
2. NEW MODE OF OPERATION
3. LOGOUT

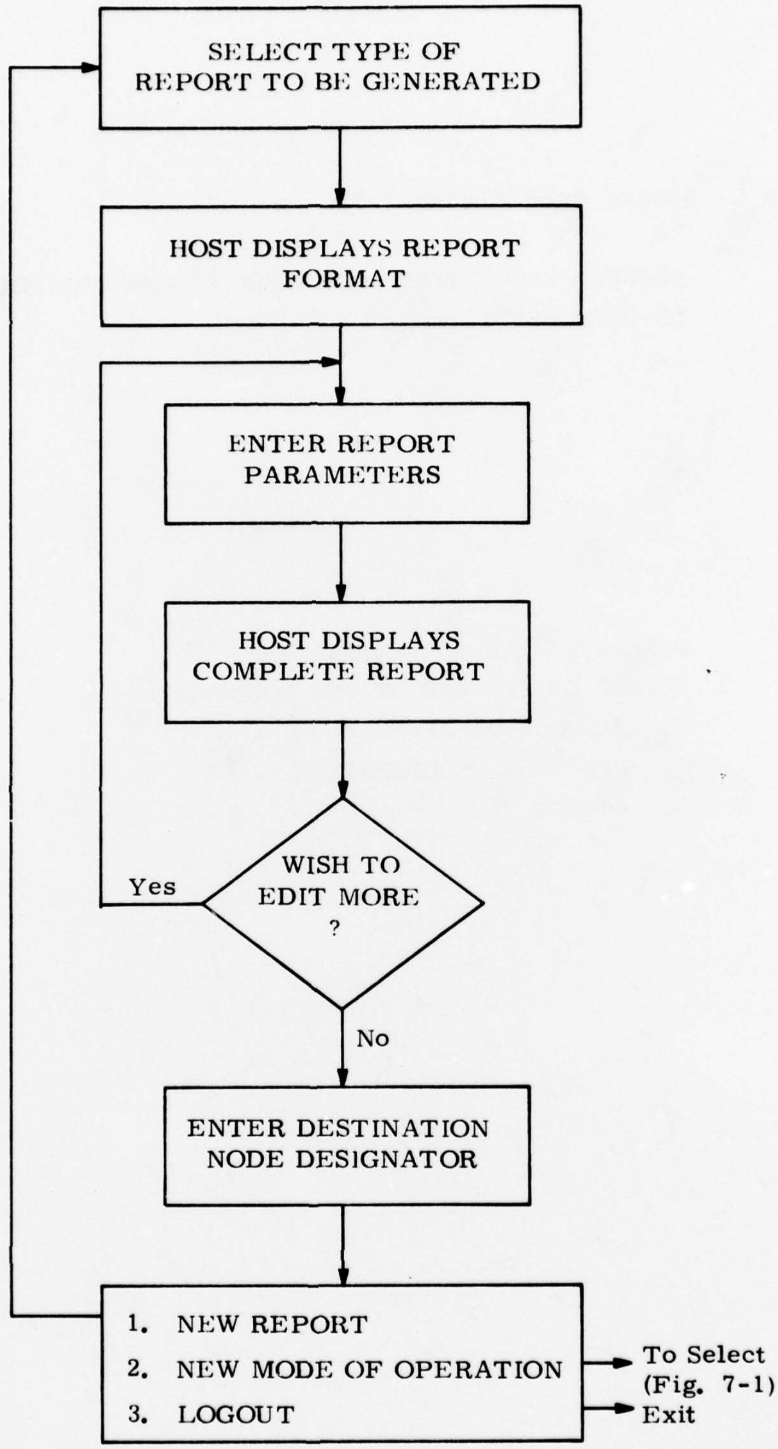


Figure 7-6
Report Mode of Operation

7.4.6 Mode 6, Status (see Figure 7-7)

HOST: PLEASE SELECT TYPE OF STATUS INFORMATION TO BE DISPLAYED:

1.
2.
.
.
.

CRT: 1 -

HOST: Displays Status Information

PLEASE SELECT ONE OF THE FOLLOWING

1. NEW STATUS DISPLAY
2. NEW MODE OF OPERATION
3. LOGOUT

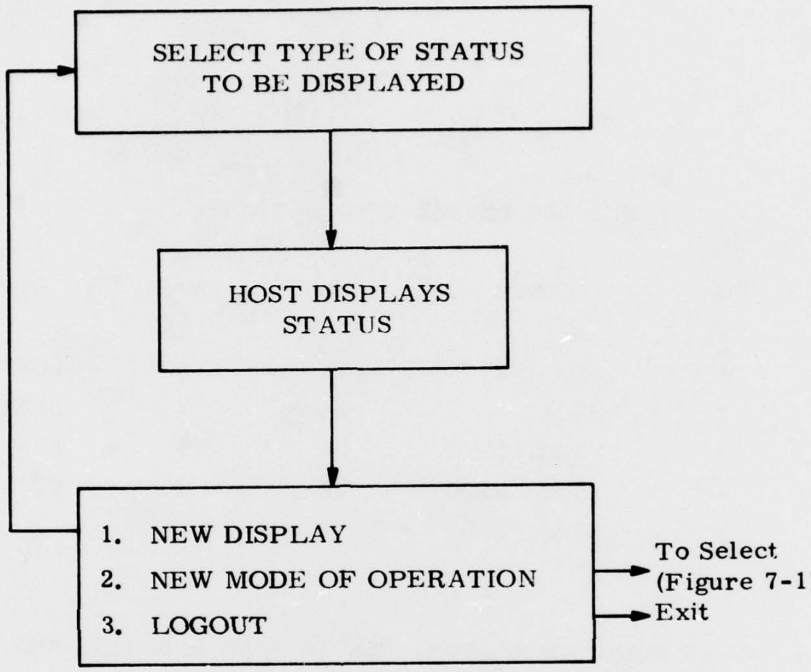


Figure 7-7
Status Mode of Operation

NETWORK DEVICE INFORMATION

ND	RDA	FUNCT	ND	RDA	FUNCT
20	1	SIM INP	25	6	OCRI
21	2	SSCI	26	7	BBSA,WBSA
22	3	VSQC, DSQC	27	8	FIAC
23	4	VSQC, DSQC	28	9	SDCA, SSCI
24	5	DBMS, PDU			

NOTE: ND IS NODE DESIGNATOR, RDA IS READ ADDRESS, AND FUNCT IS DEFAULT COLUMN FUNCTION.
PRESS CR KEY FOR NEXT INSTRUCTION

Figure 7-8
Typical Display for Network Device Information

```
              LID/FAD CONVERSION TABLE

1  1  1  4  2  1  3  3  3  3  3
0
0
0
0
```

Figure 7-9

Typical CRT Display for Logical ID's by Node (LID's 1-100)

```
0
0  0  0  0  0  0  4
0
0
0
0
0
0
```

Figure 7-10

Typical CRT Display for Logical Logical ID/Functional Address Table (LID's 101-254)

NODE WORKPAGE PARAMETERS	
OCRI NODE HAS DESIGNATOR 25, RDA 6 IN FDM.	
ALTERNATE GATEWAY FUNCTIONAL ADDRESS	2, 9
MAXIMUM INPUT QUEUE SIZE (TO EXTERNAL)	8
MAXIMUM OUTPUT QUEUE SIZE (TO BITSTREAM) . . .	1
MAXIMUM PACKET XMISSIONS BEFORE MSG TERM . .	8
TIMEOUT FOR WRITE TOKEN REGENERATION	12
TIMEOUT FOR PACKET RETRANSMISSION	41
NUMBER OF NODES IN SYSTEM.	28
NUMBER OF NODES IN FDM	9
PRESS TRANSMIT KEY FOR NEXT INSTRUCTION.	

Figure 7-11
Typical CRT Display for Node Workpage Parameters

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER 64280	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Phase I Final Report for the Modular System Control Development Model Sections 1-7		5. TYPE OF REPORT & PERIOD COVERED Sep 76 - Sep 77 Final
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s)		8. CONTRACT OR GRANT NUMBER(s) DCA 100-76-C-0083 new
9. PERFORMING ORGANIZATION NAME AND ADDRESS Burroughs Corporation Federal & Special Systems Group Paoli, PA 19301		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Task 15203 P.E. 33126
11. CONTROLLING OFFICE NAME AND ADDRESS Defense Communications Engineering Center 1860 Wiehle Avenue Reston, VA 22090		12. REPORT DATE September 1977
		13. NUMBER OF PAGES 243
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Loops, rings, system control, Defense Communication System, computer architecture

It includes an

This volume provides a

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A ~~s~~urvey and analysis of computer architectures applicable to the performance of system control functions for the ~~DCS. An~~ initial design of a modular, failsoft system utilizing a single ring network providing network, traffic, switch and transmission control functions.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)